开源时代

OPEN SOURCE TIMES

2010年2月刊 总第十七期

开源业界：

开源软件爆发 赚吆喝赔大钱

普华公司收购Turbolinux旗下的Linux核心业务

社区扫描：

75%的Linux代码经由付费开发者产生

五种开源协议的比较

行业观察：

甲骨文MySQL和Sun服务器发展计划

IBM大型机：究竟是过时还是涅槃重生？

技术新知：

开源数据库“五虎将”

基于port 基本安装 PostgreSQL 笔记

本期推荐：

Android和开源世界的故事

Linux内核开发社区宣布将Android的驱动程序从Linux内核“状态树”上除去，从此，Android与Linux开发主流将分道扬镳。后果如何？从今往后，在开源的路上，谷歌Android的硬件驱动从何而来？谷歌喜欢开源是我们所熟知的开源吗？

ChinaUnix.net

内容目录

本期编辑：周荣茂 覃里

美工：林在子

编校：王文冰

投稿信箱：rmzhou@staff.chinaunix.net

本刊网址：http：//linux.chinaunix.net/ebook/

刊首语

****

本来杂志应该在春节前发布了，因为种种原因，一致拖延到春节后才发布，在这里先给我们的用户说一下对不起！

新的一年来临了，不得不说，《开源时代》面临着种种困难，没有一个新的发展模式，纯粹来自社区的投稿和编辑团队已经显得很苍白，总是无法按时按量地完成杂志的编辑和发行任务。与此同时，杂志想朝商业化方向发展也遇到了麻烦，没有企业能够愿意站出来帮一把。

我也知道，开源不是不赚钱，可是该如何赚钱，以前总是看到各种各样的文章，教你如何如何，但是一旦你自己遇到，就不是那么简单了。一份小小的《开源时代》电子杂志就已经如此了，更何况是一个大型的软件企业?

新的一年里，《开源时代》内容方面将更加注重选题的重要性，紧紧贴合用户的需求，特别是 ChinaUnix 用户的需求。同时将结合 CU 技术沙龙和会议的内容，将其中的精华部分整理到杂志中来，给那些不能参加会议的用户一个学习和交流的机会。

除了我们的会议以外，在沙龙活动，我们将包括三个重点，开源网络和数据库运维及监控，Linux/Unix 系统管理以及开源开发技术。从 3 月份开始，我们将进行一系列的线下沙龙，包括开源虚拟化动手实验系列、双机与负载均衡应用与实践、在线存储架构优化和容灾备份架构等，在 9 月份，还有一个规模稍大一些的线上线下联动的服务器集群搭建大赛。在开发方面的沙龙，我们将联合龙芯在今年 3 月份开始启动全国范围的龙芯开发者大赛。活动将会非常精彩，到时候《开源时代》将第一时间给大家整理好相关内容并及时奉上。

送别牛年，迎来虎年，迟到的《开源时代》2010 年 2 月刊祝您春节愉快！

----ChinaUnix 社区编辑：周荣茂

开源业界

****

开源软件爆发 赚吆喝赔大钱

如果说 09 年是开源年，相信熟悉 IT 业界的人不会反对，经济危机使得开源软件势如破竹，大有舍我其谁的气概，但笔者并不认为是经济危机促进了开源软件的普及，而是积累久了，市场终于成熟的表现。

说到底，除去开源软件物美价廉的面具，它仍然只是一种商品，而不是技术人员的救世主，尽管它拥有技术人员喜欢的种种特性，但我们要知道，不是被喜欢和被崇拜就可以活的更久，Sun 作为一个备受尊敬的公司仍然逃脱不了被收购的厄运，因此我们将开源软件作为一种商品来讨论，为什么它会有今天的爆发，而在爆发中又该注意哪些市场规律。

价格因素不能成为开源软件爆发的主因

布鲁斯。亨德森：如果竞争者的条件几乎相同并以同一方式谋生，那么，他们之间的平衡就是不稳定的，这样的行业经常爆发价格战。

据市场研究公司 IDC 最近发表的一篇研究报告称，全球开源软件的销售收入从现在至 2013 年的复合年增长率为 22.4%，到 2013 年的销售收入将达到 81 亿美元，暂且不论服务器领域的主力军 Linux，单单是开源浏览器 Firefox 的市场份额就已经与 IE8 平分秋色，而甲骨文用户集团总裁兼 CEO Kaplan 曾说，MySQL 是替代甲骨文软件的最流行开源软件，根据对 269 名用户的调查，MySQL 用户占 33%.然后是 PostgreSQL 用户。

尽管开源软件的市场份额看似很大，但我们不得不注意的问题是，产品的重复度很高，而谋生的商业模式也相似，例如在中国企业级 Linux 市场上，尽管红帽的使用十分普遍，但非商业版本的使用也是占据了相当一部分市场份额，而造成这一结果的直接原因就是用户对价格的敏感。如果直观的解释，将开源软件和非开源软件划分为两个阵营，那么当开源软件吸引用户的主要原因是低廉的价格的时候，那么选择开源软件的用户多数会遵循这样的选择规律，能用免费的，不用收费的，能

用低收费的，不用高收费的，例如红帽与 CentOS 的同质竞争关系，而这样造成的直接后果就是开源阵营内部本身的价格战，势必消耗彼此的实力。因此如果我们将开源软件的爆发归于经济危机，那么开源软件本身就沦为了一种不得已的替代品，这样不仅低估了开源软件的实力，也会让开源软件厂商的日子更辛苦，毕竟拼价格不是长久之计。

价格确是成为开源软件爆发的主因

布鲁斯.亨德森：如果只有一个关键性的因素，那么，竞争就是不稳定的。在这种行业中，任何取得成本突破的公司都可以降低价格，并在损害哪些付出很大代价来保卫其市场份额的公司的利益的情况下赢得市场份额。

两个标题似乎相悖，但其实并不冲突。开源软件吸引用户的元素很多，例如更优质的服务，操纵感更强，技术更新更迅速，当然价格低廉也是其中一个元素，在这里我要强调的是，如果开源软件仅仅将价格低廉作为噱头吸引消费者，那么吃力不讨好，顶多是廉价的“神舟电脑”式营销(此处并无诋毁之意)，但是如果在开源模式下，获得成本突破，在技术优先的前提下，降低了价格，才能更加凸显其竞争优势，这就像是商店打折，本来 1000 块的东西，我打个 5 折，品质相同，价格实惠，用户当然愿意用，但如果我们将物品本身定价为 500 块，哪怕是相同的东西，用户心理恐怕也会有落差吧。

开源软件，低价不是出路

所以每每看到文章将开源软件的爆发归于经济危机，成本考虑这些因素，笔者就很替开源厂商们紧张，价格作为最后的营销手段，当开源厂商拿着自己的优秀产品面对用户“永远不够便宜”的心理一降再降的时候，哪怕再优秀的商业模式也不足以应对成本考验，那么我们是不是得到的服务越来越少，问题越来越多，而最终导致大家直接对开源怀疑，让开源死掉呢？当然了，这只是极端的考虑，随着用户群体的成熟和开源软件的成熟，开源精神塑造的不是免费的形象，而是一种优质服务，操纵和技术尖端的形象，那么到那个时候，开源软件的黄金时代才真正到来。

甲骨文将关闭 Sun 开源软件项目托管网站

2 月 4 日消息，据媒体报道，在与 Sun 微系统公司合并之后，甲骨文将停止继续访问 Sun 开发的开源软件项目托管网站“Project Kenai”。

甲骨文星期二在更新的开发人员常见问题解答声明中说，Kenai 网站将停止公开使用。甲骨文将继续在内部使用这个网站并且寻求一些让我们的客户利用这个网站的方法。

Project Kenai 团队在网站上发表的有关这个网站的前途的公告中说，目前正在逐步取消这个网站以便整合这个项目托管网站。减少当前项目托管网站的数量是朝着这个方向发展的开始。

在 kenai 测试版网站，用户被告知要把资料库和内容转移到其它地方。Kenai 团队说，这个网站的完全关闭和删除这个域名将在未来的 60 天内完成(2010 年 4 月 2 日)。这为所有的项目迁移到项目拥有者选择的新家提供了充裕的时间。在 60 天期限(2010 年 4 月 2 日)之后，任何留在该网站的项目都将被删除，这个网站将关闭。

Kenai 团队说，虽然关闭 Kenai.com 域名的时间不远了，但是，这个基础设施将来会继续存在以支持其它的域名。Netbeans.org 已经在使用这个基础设施。

甲骨文还在常见问题解答中宣布要把甲骨文技术网络、Sun BigAdmin 系统管理门户网站和包括 java.sun.com 网站在内的 Sun 开发人员网站合并在一起。甲骨文说，这个合并将产生最大的和种

类最多的开发人员、数据库管理员、系统管理员和设计师的社区。

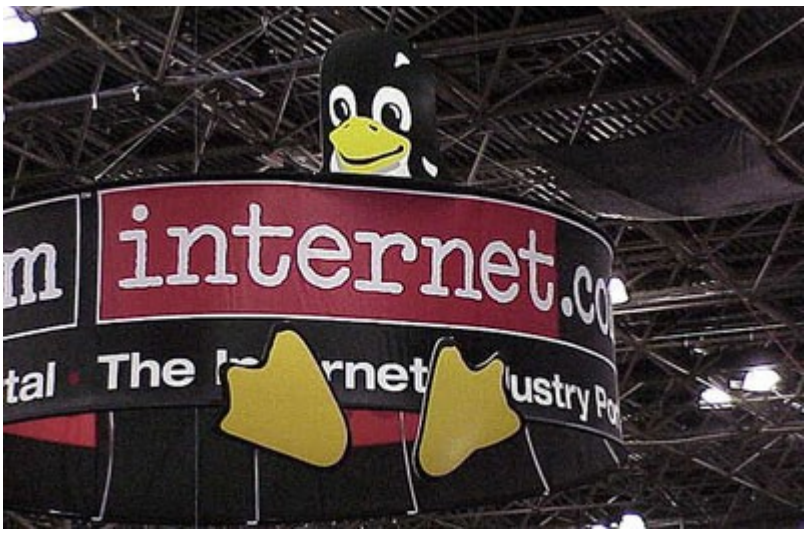
甲骨文说，在最近的将来，Sun 开发人员网络和 BigAdmin 网站将保持现状。甲骨文预计这些网站将集成为一个重新设计的甲骨文技术网络。

2010 年开源软件和 Linux 即将加冕

2010 年，围绕开源软件和 Linux 的最大预测可以说是大多数最终用户都不会谈论它，甚至都不考虑这个问题。但是，这并不是丧钟，这是一个加冕礼。

专家们不再说开源软件和 Linux 相对于衣冠楚楚的包装软件而言是胡须肮脏的局外人。开源软件正在努力进入一切领域，在智能手机等新的客户接入设备的后台发挥大脑的作用，用于采用有的开源软件技术制作的应用程序中以及在云计算中的基于服务的应用程序。这个重点是功能，不是基础的技术。

Sleepy Cat 公司创始人、现在的 Cloudera 公司首席执行官 2009 年年初对 Network World 网站说，在 Sleepy Cat，我们为我们是一家开源软件公司而自豪。在 Cloudera，我们把自己当作是碰巧在开源软件的基础上建立起来的一个企业软件公司。



此外，政府和其它 IT 机构在 2010 年将通过基于现行政策的应用确认开源软件，并且开始根据特点和功能的需求从正门引进开源软件，而不是从后门悄悄地引进开源软件。市场研究公司 IDC 在讨论其 2010 年预测的技术部分的时候省略了“开源软件”这个词汇。其主要原因是承认开源软件的功能，而不是开发模式。

IDC 负责系统软件的计划经理 Al Gillen 说，我考虑了开源软件为什么已经成为这个行业的一个标准部分的原因。开源软件不是我们表示不满的东西。目前正在发生的一些更大的趋势有一部分或者以显著的方式是由开源软件帮助、推动或者实现的。这下面的开源软件是我们预测在 2010 年将发生的事情的一个非常重要的组件。

Gillen 说，这个事实在云计算方面是最突出的。他说，对于正在建设一个不是从目前安装的应用程序派生的云计算服务的任何人来说，Linux（和/或者）开源软件组件很有可能成为部署在这个云计算的软件栈的一部分。今天，你没有任何其它选择。你不能建设一个 Windows 云计算。

微软在 2 月份将开始为其 Azure 云计算平台计费。但是，这里的主要推动因素是集合开发人员

制作人们要消费或者生活离不开的应用程序。据内部人士说，对于开源软件来说，“平台”还将构成 2010 年的另一个大的主题。

MindTouch 首席执行官 Aaron Fulkerson 说，开源软件应用程序正在成为一个正在建造之中的平台。这使许多正在考虑制作应用程序的开源软件厂商感到意外。Fulkerson 说，随着基本的软件包成为新一类应用程序和服务的基础，这种趋势在 2010 年将继续下去。

451 Group 分析师 Jay Lyman 说，当企业评估新的工具的时候，开源软件和 Linux 在 2010 年将获得与专有软件相同的地位。他引述了 2009 年 10 月发表的美国国防部备忘录。这个备忘录澄清了美国国防部使用开源软件的指南。这个备忘录说，开源软件技术几乎在所有的方面都与商业软件相当。根据法律规定，美国国防部在做出采购决策的时候应该考虑开源软件。

Lyman 说，开源软件的经济、发展和成熟以及更多的商业技术支持正在使企业关注开源软件技术。他说，我们仍将看到在许多机构确实对开源软件应用的政策。但是，像美国国防部备忘录这样的事情将推动更多的官方的和正式的应用开源软件。

Lyman 说，其它预测的 2010 年的趋势包括开源软件在健康医疗等新行业中的更广泛的普及。他举例说明了 Medsphere 公司围绕 OpenVista 电子健康记录软件提供的技术支持选择。他还认为开发人员和运营经理在开源软件方面的界限正在模糊起来，因为这两组人员在会议上和有关项目的讨论中都是肩并肩地坐在一起。厂商更可能在自己的开源软件的努力中解决基础问题和其它小组的问题。他说，这是针对开源软件的“基础的方法”。

他补充说，2010 年另一个重要的趋势是中小企业更多的采用开源软件。有更多的渠道运营商和增值转销商、系统集成商、托管服务提供商和其它服务提供商。所有这些运营商都将把更多的开源软件集成到自己的产品中。因此，在许多情况下，中小企业客户确实不知道这是开源软件或者他们不在意是什么软件。

Linux 基金会执行理事长 Jim Zemlin 说，日益增值的免费硬件概念在 2010 年将爆炸式增长。他说，这是一个正在朝着一种服务模式发展的行业。你在那里能够从 iTunes 或者应用程序商店、或者诺基亚的 Ovi、或者无线数据计划等服务等在线服务中赚钱，而不是依靠硬件赚钱。你将看到更多的与服务捆绑在一起的免费的计算机、免费的手机和免费的消费者产品。

Zemlin 根据这些线索预测说，应用程序仓库将在 2010 年崛起。这个地方将给开发商一个供应链以便把他们的应用程序发布到许多销售点。那可能是一个运营商、一个操作系统厂商或者一个 PC 厂商。但是，一个独立的软件厂商将不会把自己的软件产品销售到没一个单个的应用程序商店。Zemlin 还说，有关 Linux 在上网本上已经死亡的报道是言过其实的。他说，Linux 基金会正在考虑一些事情。但是，他不愿意讨论这些事情。

总之，2010 年将出现各种各样的许多发展和进化的步骤，其中再一次包括 Linux 桌面，特别是由于 GNOME 将以在 9 月份推出 3.0 版的方式进行重大的转变。重大的改变不是 Gnome 的 DNA 的一部分。3.0 版将推出一种使用 Clutter 工具制作的新的桌面外壳并且增加一个名为“Zeitgeist”的新工具。这是一个替代传统的文件管理方法的日志式的工具。

Novell 的 openSUSE 社区管理员 Joe "Zonker" Brockmeier 说，2010 年还可能是一个令人大吃一惊的一年。他说，2009 年显然是一个非常艰难的一年，有许多人被解雇。因此，一些聪明人发现自己失业了并不让人感到意外。许多聪明人有许多闲暇时间。

Novell 低调转型 借三股“东风”谋划云计算

2009 年 12 月初，Novell 推出了最新的智能工作负载管理战略和产品规划蓝图。这是 Novell 因应客户在安全管理和优化其分布式计算资源方面的需求而提出的最新解决方案。“其实，IWM 就是云计算解决方案，与其以陌生的词汇来一点一点给客户普及，不如以最直接最通俗易懂的方式来让客户了解”，Novell 全球副总裁、东亚区总裁张先民在接受专访时说道，“也就是说，Novell 正式进入云领域。”

Novell 全球副总裁、东亚区总裁张先民博士

三股东风“吹动”云

云计算是近几年新兴的一项非常热门的技术，IDC、Gartener 均将其视为未来必然的技术趋势。之所以能将这个“云”在短短时间内吹得如此声势壮大，张先民总结主要是以下三股“东风”的推动：

第一股东风即科技推动。云计算有两个非常重要的科技，一是网络、带宽现在非常强大，无线也非常普遍。二是虚拟化技术很成熟，在技术上已经能够支撑云计算。第二股东风即经济诱因，金融危机使很多企业受到了严重的损失，公司都要节省成本。第三股东风即政治导向，现在最热门的话题就是气候变迁，“节能减排”已经不再只是口号，能耗能减少多少是很重要的指标。比如中国电信业，中国移动，中国联通和中国电信都在很严肃地研究这个课题。

这三股东风一齐吹动，势必将为云计算带来一个爆发性的成长。而 2010 年就是云计算开始大爆发的起点，张先民如是断言。

五年筹备终入云

云计算前景虽好，可要想从中分一杯羹，也并不是谁都可以的。Novell 作为在 Linux 领域颇有建树的基础软件提供商，直到 2009 年底才低调宣布进入云领域，这之前也是经过了五年左右时间的准备和筹划的。

从基础架构最底层构建方面的虚拟平台 SUSE Linux Enterprise Server 和 SUESE Studio 等，到管理方面的 Platespin Orchestrate，安全方面的 Sentinel 以及优化方面的 BSCM/BSLM 等多款软件，再到用户门户方面的 Atlantic，Novell 已经通过这五年多来不断的自主研发和对外收购完成了自身在云计算基础架构软件工具提供商的初步布局。

张先民透露，今后一年，Novell 还将推出 8 款产品——包括 SUSE Appliance Toolkit、Identity Manager 4、Workbench 等——完善和扩展 Novell 在云计算市场的产品领域。这些产品将解决用户在整个生命周期中遇到的在创建、保护、管理和衡量方面的困难。并且，由于产品采用模块化的方法，用户可以从任何产品的生命周期的任何时候开始，并确信他们正开始走上云计算之路。

招兵买马迎接云爆发

2009 年 11 月，在采访前 Novell CTO Jeff Jaffe 时咨询过美国的云计算应用情况，当时得到的回答仅仅是 1%。1 个多月后的今天已经跨入 2010 年，若真如张先民博士所言，云计算将迎来爆发性增长，中国市场那将是何等繁荣呢?

在张先民看来，不用去预测，光从现在中国地区应接不暇的招标案来看，2010 年对于 Novell 来说就将是一个繁忙而收获颇丰的一年。从当前市场来看，电信运营商在云方面的需求和市场是最旺盛的，因此 Novell 2010 年将把电信业作为关注的重点。

对于中国电信业市场，张先民分析道：“客观预计，2010 年三大运营商在云部署方面不会特别快。中国 IT 使用本来就在美国之后，每个省份发展也不一样。要上升到什么程度，本身也有一定的制约因素，不会一下子从物理机就到云，总要慢慢试验。但是明年年底如果有一半以上省份开始做云，我并不觉得惊讶。而且大规模接触云这个不难，但是想成熟应用，我想还要 2、3 年甚至更久。”目前，电信运营商最希望的就是把自己的 IDC 云化。因此，Novell 正与中兴等多家系统集成商合作，一起为电信业者建立一个带有云特色的 IDC 而努力。

张先民还透露，由于人手不足，对于来自全国各地的招标案已经应接不暇，另外也是为了迎接即将爆发的云计算应用，Novell2010 年将开始招兵买马，将现有的百人团队扩大到到 130—150 个人。招募人员不仅局限在销售、市场人员，专业的开发人员更是 Novell 所紧缺的。

Vmware 宣布收购雅虎 Zimbra 开源消息业务

VMware 近日称，它将收购雅虎的 Zimbra 开源软件消息业务以增强其正在发展的协作平台。

这笔收购交易的条款没有披露。但是，各种消息来源报道称，这个销售价格接近 1 亿美元。雅虎在 2007 年用 3.50 亿美元收购了 Zimbra。

公司官员称，VMware 已经收购了与 Zimbra 有关的全部知识产权。Zimbra 是在 Linux 和 MacOs 平台上流行的开源软件消息平台，在大学和服务提供商中也很流行。雅虎仍保留使用这个产品的一些部分的权利。这些部分已经建在雅虎邮件和日历等产品中。Zimbra 拥有 5500 万邮箱，

2009 年的增长率是 86%。

Burton Group 分析师 Chris Wolf 说，这笔收购交易对于 VMware 是有意义的。VMware 认识到它需要从虚拟化的重点向其它方面扩展，因为微软等竞争对手威胁要大举进入虚拟化市场。收购 Zimbra 对于 XMware 来说是一个非常明智的举措。这是重新定义传统的应用程序栈和企业预计从中得到什么的努力的一部分。VMware 副总裁兼负责云计算服务的总经理 Brian Byun 说，Zimbra 是伸缩性的“云计算时代”解决方案的一个极好的例子。这种解决方案能够把较小的现场实施扩展到云计算。它将是扩展解决方案产品组合的一个构件。这些解决方案可以作为一种虚拟设备提供，或者由服务提供商提供。

在 2009 年 4 月，VMware(现在由前微软官员 Paul Maritz 经营)收购了 Java 厂商 SpringSource。这个公司制作 Web 应用程序开发和管理工具，重点是在开源软件方面。这次收购首次表明 VMware 在努力为其虚拟化专业知识增加新的技术。

中国企业青睐 Android 紧跟“巨头”不掉队

两年时间里，由谷歌公司主导成立的开放手机联盟（OHA）成员数量迅速攀升，其中也出现了很多来自中国大陆和台湾地区的企业身影。截至目前，国内三大运营商、中兴通讯和华为等厂商以及来自产业链细小领域的业务提供商都已经加入该联盟，据统计，OHA 的中国成员数量已达到 12 家，占据总成员席位的近 1/5。

“应该说，谷歌公司的强大号召力在中国得到了很好体现，前瞻的技术、可靠的价值链、持续的创新意识让中国很多企业一致看好 Android。”GartnerIT 系统高级分析师沈哲怡表示。

广东移动数据部人士也表示，“Android 在实践‘免费’经营模式的同时，也为移动终端领域的后进入者打开了一扇大门，彻底的开源和免费无疑是降低了市场进入门槛。”

市场挑战者的瓶颈

从产业发展角度看，Android 产业在中国的迅速蔓延本质上也存在一定的市场需求。在加入 OHA 的中国成员中，无论是终端厂商还是业务提供商，在终端领域的市场份额并不占据优势，而国内运营商对手机定制的紧密程度也与国外相差甚远，因此中国成员大部分都充当了全球市场挑战者的角色。另外，还有一个群体值得关注——山寨手机产业，尽管其身份还有颇多质疑，但从规模和经济收益角度，中国的山寨产业已经成为手机市场不可忽视的“潜在力量”，但受制于成本和正版的限制，手机山寨产业对于手机操作系统的渴求更加强烈。

但无论是对正牌或山寨的手机制造商、业务提供商还是电信运营商而言，他们的瓶颈都在于“拥有自主开发的手机操作平台并非易事”，即便是处于产业核心层面的电信运营商也不例外。

一位从事智能手机应用软件开发的杨姓项目经理告诉记者，开发手机操作平台是一项庞大的工程，“靠公司规模和资金实力并不能解决这一难题，这项工程考验的是企业对网络和 IT 技术的理解力和创造力”。

而谷歌适时推出的开源、免费的手机操作平台 Android 正好满足了这一市场需求，为产业链相关企业提供了延伸业务领域的可能性，更为移动互联网领域的后进入者提供了开拓市场的有力武器。

OHA 的成员可以任意使用和修改 Android 的 SDK 文档进行再开发，由此形成适应自身业务需要的“定制版本”。Android 的这种开源特征很快得到了中国运营商的响应，中国移动、中国联通、中国电信相继加入 OHA 联盟。“在 OPhone 的研发中，中国移动对 Android 平台的上层应用、界面风格以及按键设置进行了大量修改，替换了很多与中国移动自身增值业务相冲突的业务应用，使中国移动的自身属性得以体现，这是 Android 的开放性所致，也是运营商选择 Android 的理由。”前述广东移动人士表示。据了解，中国联通基于 Android 平台的 Uphone 也在紧张研发当中。

“值得国内厂商信赖”

“Android 的技术结构比较紧凑，市场推进有张有驰，再加上谷歌公司长期具备的创新精神，确实值得国内很多厂商信赖。”前述分析师沈哲怡表示。

据了解，国内终端厂商和运营商看好 Android，很大程度上也是因为其背后推动者谷歌对互联网趋势的深刻见解；同时也因为 Android 具备合理的系统内核设计，并拥有成熟的市场推广模式。

而在这方面，同样是倡导“开放”的开源平台 Linux，其市场表现却与 Android 有着天壤之别。截至 2009 年，Linux 应用在桌面操作系统中只占据 1%的市场份额，始终难成操作系统的主流。

沈哲怡对此表示，仅仅是技术开源还不够，如果操作系统开发企业缺乏必要的市场化运作和整体推动力，忽视用户界面体验，最终其产品将无人问津，而 Android 在此方面则是一个成功案例。

据了解，谷歌近年来还积极推进版本更新，不断完善其系统性能，在 2010 年 1 月 13 日，谷歌公布了 Android2.1 版本的 SDK 文档，也就是说开发人员甚至可以对谷歌自有品牌手机 NexusOne 进行修改和业务程序开发，有外媒评论，“此举真正体现了 Android 平台的开放和平等”。

紧跟“巨头”不掉队

而在 Android 的设计理念中，开放和平等也有所体现。“开放性接口为应用程序的研发提供了一个宽阔平台，业务提供商不必担心技术垄断造成的压迫和威胁。”前述杨姓项目经理表示，其实 Android 的很多底层技术取自 Linux，而在底层技术流的选择上，大部分的程序开发人员骨子里都有一种“反 Windows”观念，“如果让一种技术处于长期垄断地位是十分可怕的，垄断者的轻微举动会对产业上下游造成很大影响，因此大部分技术人员会更倾向于开放的 Android。”

而从系统整体性角度，Android 平台也提供了完备的解决方案，OHA 的成员甚至可以拿来直接使用。如此讨巧的平台设计在中国迅速聚拢了一批通信领域的“明星企业”，像华为、中兴这类专注于通信领域，但已具备 IC（集成电路）能力的厂商能够凭借此平台快速切入终端领域，降低成本且缩短了研发时间。

巨头的选择让 OHA 在中国的发展更加明朗——华为、中兴、联想等国内终端厂商以及国内三大运营商的加盟，让很多产业相关环节的厂商看到了趋势，于是，代工厂商、芯片厂商以及手机配套应用厂商源源不断地加入其中。

“很明显，谷歌的开放模式既然得到了产业链主要成员的认可，那些不具有话语权的小厂商当然也会紧紧跟随，它们认为‘巨头’往往代表产业发展趋势，如果不及时跟进，难免就会掉队。”前述杨姓项目经理表示，比如来自中国台湾地区的联发科、宏基、威盛、富士康、华硕都已经成为了 OHA 的成员，“中国台湾地区属于电子产品的高密地区，拥有很多手机代工商和芯片制造商，因此这种特征也尤为明显”。

微软：Linux 将成为手机操作系统洗牌输家

微软手机业务掌门罗比·巴奇(Robbie Bach)称，手机操作系统领域将会洗牌，Linux 将成为输家。巴奇预测，Linux 手机无法通过“质量”测试，不会在市场上获得成功。他说，Linux 手机操作

系统版本多达 17 种，移动运营商不会支持如此多的操作系统。尽管大多数 Linux 手机都是低端手机，但最近部分手机厂商推出了 Linux 智能手机，其中包括索尼爱立信。谷歌 Android 也是一款基于 Linux 的操作系统。部分企业客户选择 Linux 智能手机，并将此作为在企业中使用开放源代码软件战略的一部分。

巴奇预测，移动运营商将拒绝部分 Linux 手机操作系统版本，但没有具体指出是哪几种版本，因

为过多的版本会使技术支持工作复杂化。

业内人士指出，最近 Windows Mobile 市场份额出现起伏，将受到 Android、iPhone 和黑莓的冲击。据悉，Windows Mobile 7 将于今年晚些时候或明年发布，竞争对手将有更多时间扩大他们的领先优势，这使得 Windows Mobile 前途更令人堪忧。

巴奇上周在国际消费电子展上表示，“我认为，除了用户体验对企业用户没有吸引力和不够现代外，Windows Mobile 并没有面临具体的挑战。”

巴奇说，“我认为目前手机领域的操作系统已经太多了，低端手机领域就有 17 种 Linux 版本，每个版本要求不同的网络认证、支持。部分 Linux 版本将被淘汰，原因不在于没有市场空间，而在于他们在质量和规模方面有所欠缺。”他指出，微软的任务是确保 Windows Mobile 有足够的规模，但没有预测洗牌后能够生存下来的手机操作系统数量，“我们肯定将成为赢家之一”。

Linux 稳步成长 5 年内相关工作岗位增多 80%

Linux 基金会表示，在过去 5 年里，与 Linux 相关的工作岗位增多了 80%。为了适应并推动这种趋势，Linux 基金会发布了一个 Linux 工作台，为 Linux 相关工作求职者和雇主提供一个交流的平台。

Linux 基金会去年从 GeekNet 那里购买了 Linux.com 域名，通过该网站为 Linux 用户和开发人员提供各种各样的内容和服务，包括是博客和社区技术支持。此次上线的工作台是 Linux.com 的最新上线内容。

Linux 基金会执行理事 Jim Zemlin 在一份声明中表示：“Linux 在行业中的使用日益广泛，这也提供了日益增多的 Linux 工作。Linux.com 的读者包括成千上万的全球 Linux 专业人员。通过在社区里提供一个工作台功能，我们可以将雇主、求职者和招聘人员联系到一起，为更加美好的 IT 业明天贡献一份力量。”

2010 展望：Unix 期待未来新标准

在过去的 40 年里，Unix 操作系统帮助推动了全球的重要任务 IT 运营。现在，随着 Unix 步入中年，它的支持者正在忙于开发新的技术规范。他们希望这些新的技术规范将推动这个操作系统进入下一个计算时代。

在 Unix 生态系统最前沿的是 Open Group，这是一个厂商中立和技术中立的组织，负责监管 Unix 名称的使用和遵守 Unix 的技术规范，特别是“单一 Unix 技术规范”。单一 Unix 技术规范是构成兼容 Unix 系统的一套技术规范。目前，单一 Unix 技术规范是第三版(Unix03)，不过，新的技术规范正在研制之中以扩展 Unix03 的成功。OpenGroup 的标准总监 Andrew Josey 说，我们的 Unix03 应用的增长感到非常满意。我们正在研制这个标准的进化版本。我们现在已经完成了基本的技术规范。我们目前正在研究如何推出这个版本。

Josey 说，这样做不包括重复发明，而是预示着 Unix 将发生革命性的变化。任何新的技术规范在性质上都是革命性的，特别是将继续支持现有平台和以前的 Unix 技术规范。那是非常重要的，因为向下兼容性一直是 Unix 的一个重要的特色。据 Josey 说，符合 1995 年的 Unix 技术规范的应用程序二进制代码在目前的平台上仍然可以使用。这种可靠性已经成为 Unix 部署和提供基于这个操作系统的系统的许多厂商的一个关键的卖点。兼容 Unix03 的操作系统来自许多厂商，包括惠普、IBM、Sun、甚至苹果。苹果自从 2007 年以来一直兼容 Unix。Josey 说，我们的厂商一直是非常保守的。

保证继续向下兼容性的措施并不是支持者希望的下一个版本的 Unix 将发光的唯一的地方。Josey 说，我们将提供一些新的编译库功能，让应用程序编程接口更加丰富。我们将期待着更好的国际化的支持、多线程的支持、更加强大以及更好的线程处理。

Unix 10?

虽然测试套装软件现在已经为下一个版本的 Unix 技术规范制作好了，但是，在新的 Unix 技术规范正式获得批准之前，仍然有更多步骤的事情要做。Josey 解释说，Open Group 首先制作了一套反应新的技术规范的更高水平的标准。然后，这些标准将经过一个委员会的审议阶段以帮助建立共识。最后，Open Group 的成员将执行一个正式投票过程以批准最终的技术规范。

虽然新的 Unix 技术规范将建在 Unix 03 标准的基础之上，但是，目前还不清楚新的 Unix 标准实际上叫什么名字。Josey 说，我们这一次也许没有做一个品牌编号。我们正在争论如何包装这个软件。如果我们确实要给它一个编号，我们也许把它称作“Unix 10”。不过，我们也许不会叫这个名字。

Linux 的挑战

虽然 Unix 在它存在的 40 年里经历了各种各样的挑战，在市场上威胁 Unix 地位的、最近的和持续的威胁之一一直是 Linux 的兴起。然而，尽管 Linux 有吸引力，特别是在数据中心，但是，Open Group 认为 Linux 并不是 Unix 或者 Unix 标准的一个威胁。相反，这两个操作系统之间的关系已经成为互补的。

Josey 说，我一直把 Linux 看作是一种非常积极的东西。我们一直在与这个团体合作。我已经看到了 Linux 从不兼容 Unix 到更兼容 Unix 的一些的变化。Unix 厂商和 Linux 厂商之间的界限最近几年已经变得非常模糊了。包括惠普和 IBM 在内的主要 Unix 厂商现在也是 Linux 的主要支持者。

Open Group 首席执行官 Allen Brown 说，我们的成员包括在 Unix 和 Linux 的供应商。这不是说一个比另一个更好。他们有不同的目标和不同的功能。Brown 说，尽管 Linux 增长并且缩小了与 Unix 操作系统与各种版本的 Linux 之间的区别，由于一些重要的原因，Unix 在未来若干年里仍然继续是一个重要的平台。

Brown 说，40 年前，像我们这样的一些人发布了我们认为不会持续很长时间并且可能被淘汰的应用程序。现在，这些应用程序是我们不能抛弃的并且必须与其它东西集成在一起的遗产的一部分。

Josey 补充说，Unix 在美国军用飞机等资产中的应用已经持续了 50 多年。因此，他希望 Unix 将继续充满生气和活力。Brown 说，与其它平台不同，Unix 越来越多地在重要任务环境中应用。因为他们是重要任务的部署，一旦应用就很难把它们从基础设施中消除。

甲骨文收购 Sun 后所面临的 4 个巨大挑战

随着欧盟在上周四终于做出批准决议，甲骨文公司斥资 74 亿美元收购 SUN 公司的交易历时 9 个月的努力几乎已成定局（中国和俄罗斯当局的批准目前还在进行当中）。

但现在的难题是：这是市值 110 亿美元的硬件及软件厂商与市值 230 亿美元的软件巨人之间运营模式的结合。甲骨文需要花大力气去理清这中间的头绪。

以下是甲骨文今后几个月里可能面临的四大挑战：

1、止血

去年 9 月，甲骨文的首席执行官拉里.埃利森评论说，由于计划内的收购被拖延，SUN 每个月都会损失 1 亿美元。增值经销商和分销商都表示，由于甲骨文可能会停止那些 SUN 的产品，哪些在收购后会继续保留带有很大的不确定性，目前用户对购买 SUN 服务器持观望态度。根据 Sun 公司 9 月 27 号结束的第一财政季度报告显示，SUN 公司销售额与去年同期相比下降了 33%。

解决这个问题的第一步是对甲骨文打算继续保留那些产品和将要淘汰那些产品的大量细节进行快速沟通。埃利森已经许诺会继续（甚至增加）正在研发当中的 Sun Sparc 硬件，Solaris 操作系统，JAVA 开发平台和开源 MySQL 数据库上的投资。不过埃利森也表示，甲骨文公司也把关注度集中在"高价值，高性能"计算机市场，并且无意在商用服务器市场上与戴尔和惠普公司进行竞争。这就留下了某些低端服务器和存储产品被放弃的可能性。

甲骨文公司计划在下周三进行时长 5 个小时的网上直播，公布公司并购后的产品路线规划图。这必须是一份非常详细的通报（不是市场宣传），同时还会涉及产品终止计划，不适合公司长期发展战略的岗位裁撤等。

2、说服客户和合作伙伴，甲骨文公司对硬件产品忧心忡忡

本次收购一拖再拖，有关甲骨文打算将 SUN 硬件业务出售给另一家公司（富士通是最频繁出现的假设）的传闻也时不时的流传着-尽管 Sun 公司的 Solaris 操作系统，MySQL 数据库，Java 开发平台及其他软件产品都是实打实的明星产品也无济于事。

这个问题与 SUN 的财务危机息息相关。解决方案也跟这个密不可分。目前甲骨文公司唯一的硬件产品是甲骨文和 SUN 合作并使用公司的 Flashfire 技术共同开发的 Exadata Database Machine，数据库机时建立在 Oracle Exadata Storage Server 上的。

埃利森和甲骨文其他高管都曾探讨过利用合并后的 Sun 硬件和甲骨文软件来创建即插即用设备，现在是提供甲骨文公司即将开发和销售的硬件/软件组合细节的时候了。目前甲骨文已经拥有了作为模型的 Exadata 系统-这是一个良好的开端。

3、让世界接受开源软件

诚然，甲骨文公司也拥有，销售和放弃过某些开源产品，诸如 Berkeley DB 数据库和 Innobase 事务性存储引擎等产品。但正如 Forrester 前任分析师王志强在收购公布后不久发表的分析报告所言，收购 Sun 公司将使得甲骨文成为开源行业某些宝贵财产的守护神，其中包括常用的 MySQL 数据库，

Solaris 操作系统和 Java 开发平台。

无论从市场营销还是销售的角度来看，甲骨文是一个非常有竞争力的公司都是无可争议的事实，可以毫不犹豫的和 IBM(纽约证券交易所代码：IBM)和 SAP(纽约证券交易所代码：SAP)这样的竞争对手共同角逐 IT 蛋糕。但问题是，如果有机会的话，甲骨文是否能调整并利用甲骨文公司具备竞争优势的开源技术。甲骨文能否抵御不利的一面，只吸收好的一面呢?

4、两大 IT 巨头合并后具体细节的管理

甲骨文面临着将 SUN 的各种硬件和软件和自己的运营模式相融合的任务，同时还要保持甲骨文公司的运营高利润率。市值 230 亿美元的公司与市值 110 亿美元的公司之间的结合将意味着大量的领土之争，以及管理层之间的冲突和员工士气，企业文化的各种不确定性。

这都不是容易的事。对甲骨文来说幸运的是，在过去 5 年中对仁科，Siebel Systems 和 BEA 系统公司等大型公司的收购帮助甲骨文积累了两家企业合并后企业运营融合的丰富经验。在 SUN 收购仁科时，许多观察家预计会发生很多文化冲突和其他困难。但甲骨文坚持下来了。该公司能否再次成功呢?

普华公司收购 Turbolinux 旗下的 Linux 核心业务

一个类似“联想收购 IBM”的案例正在基础软件领域上演。昨天，国内最大的基础软件企业——普华基础软件宣布，已经对全球著名的 Turbolinux 公司旗下的 Linux 核心业务实现控股，并将在未来 2 年内完成 100%的收购。

这是国内基础软件厂商首次展开海外投资行动。普华公司是中国电子科技集团下属企业，总部位于上海，注册资本达到 2 亿元。

据悉，普华以现金出资，拥有了 Turbolinux 下属的专注于 Linux 技术研发的 TurboSystems 公司 51%的股份。这相当于收购了 Turbolinux 的核心业务。对于具体收购金额，普华表示不方便透露。不过，普华总经理赵晓亮告诉本报记者：“我们以一个非常合适的价格完成了收购，代价和风险并不太大。”Turbolinux 成立于 1992 年，是全球最著名的 Linux 三大品牌之一，是 Linux 软件解决方案和企业计算基础构架方面的领导者。

“我们迈出国际收购的第一步，是中国基础软件企业实力的体现。”赵晓亮表示。此前，通过开发与微软视窗操作系统竞争的 Linux 软件技术，国内已经形成了一些基础软件厂商，但是每家公司规模都不大，且大都“单打独斗”，即使有合作也很松散。不过这种局面在过去 2 年出现明显变化。随着 Linux 系统在全球影响力的日益提升，国内各类信息化领域对操作系统等自主基础软件的需求不断提升。同时，国家主管部门意识到，在基础软件领域不能再受制于人，必须加大投入，整合资源，组建旗舰企业，以此实现中国自主基础软件的突破。根据中央的“核高基”（即核心电子器件、高端芯片、基础软件）专项规划，仅“十一五”期间，中央财政就向基础软件产业投入 33 亿元，同时，地方政府还将按照 1：1 的比例配备扶持资金。估计到 2020 年，投入资金将达 100 亿元。

在这种情况下，海外收购的时机变得日益明晰起来。赵晓亮告诉记者，与 Turbolinux 的收购商谈启动于去年七八月间。此后，收购细节的谈判在两家公司之间紧锣密鼓地展开，同时也得到了商务部、工信部等的认可和支持。赵晓亮表示，此次收购主要是看重 Turbolinux 的技术和团队。收购后，普华可以借助外力，提升服务国内市场的实力；同时也可以在力所能及的情况下，试探性地朝海外市场“踏出一只脚”。

“我们收购的规模暂时还不能和联想收购案例相比。”赵晓亮告诉记者，但他承认，这次收购

能为将来展开更大规模的国际收购做一些探索。目前，国际基础软件领域已经出现了整合的趋势，其中，全球最大的数据库企业甲骨文（Oracle）通过收购 Sun 公司，补上了其在操作系统领域的短板。“现在，我们还没有收购 Sun 这样规模的企业的能力，但中国基础软件企业要想真正做强做大，未来不排除参与更大规模跨国并购的可能。”赵晓亮表示。

红帽中间件：普及 JBoss 是今年工作重点

Red Hat 的 JBoss 中间件部门会有一个忙碌的 2010 年，他们将继续为提高开发工具和 Java 服务器而不断努力。虽然性能改进总是 JBoss 的重要任务，但今年的重点将会放在帮助开发者改进开发工具和服务器的使用方法上。

在 JBoss 今年的各项工作中，重中之重是对 JBoss Developer Studio（JBDS）的改进。JBDS 即原先的 Exadel Studio Pro，开始是软件厂商 Exadel 的一个封闭源代码项目，后来和 JBoss 合作开发之后在 2007 年成为开源项目。

Red Hat 中间件首席技术官 Mark Little 在采访中说，JBDS 会在常规发布中增加许多新的功能，但前进的关键是关于怎样提高生产力。“我们已经定下了今年或今后两年的首要任务，那就是提高生产力，另外还要让我们的平台和所有的项目都能够拿来即用，”Little 说，“JBDS 是一个关键的组成部分，人们期望我们的工具能够容易上手，尤其是那些从非开源厂商转过来的用户。IBM 和 Oracle 正在这样做，我们的 JBDS 也差不多。”

Little 补充说，过去 JBDS 的重点一直放在 JBoss 应用服务器。而展望未来，JBDS 将扩展到整个 JBoss SOA（Service-Oriented Architecture）平台。“我们现在开始看到一些基于 SOA 的 JBDS 工具正在出现，而且仍处于相对初级的阶段，”Little 说，“因此，在未来一年左右的时间里，我希望看到越来越多这样的工具出现。”

另一个 JBoss 的相关增长领域是如何将 OSGi（Open Services Gateway initiative）模块化方法连接到 Java。Little 表示近来 JBoss 一直在寻求 OSGi 上的突破，但还没有投入很大的努力。“我们已经看到一些开发者把在其他容器上建立的 OSGi 包部署到 JBoss 应用服务器上，”Little 说，“很多是从 SOA 开发而来的，其中有一个 OSGi 包可以成为一个非常不错的封装服务单位。”

因此 JBoss 已经开始研究 OSGi，但并不是要取代自己的微型容器架构，而是作为一个可以协同配合的选择方案。“我们可以支持 OSGi 包与原始的格式同时运行，”Little 说。对 OSGi 的支持有一部分是通过 Red Hat 在 2009 年 JavaOne 大会上推出的 JBoss OpenChoice 方式实现的。JBoss 还宣布了一项计划，通过使用 OpenChoice 推出一个轻量级的 Java Web 平台服务器。

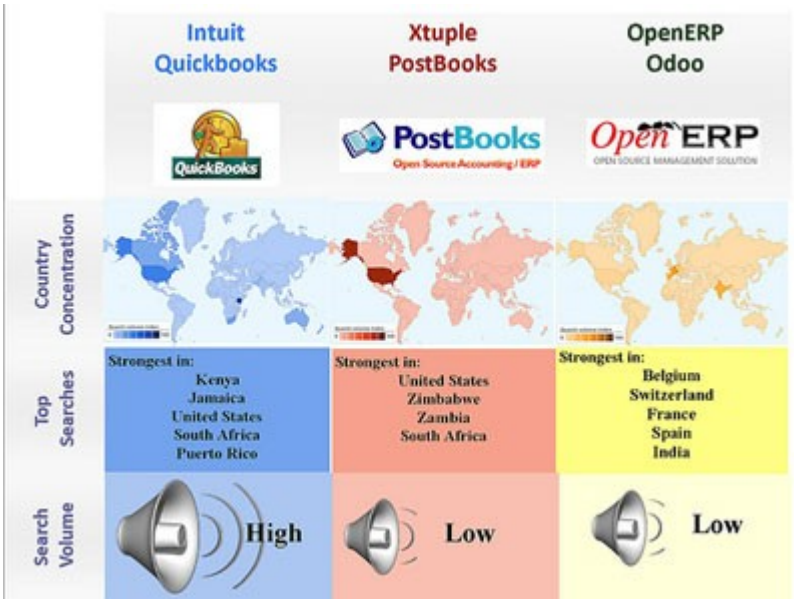
Little 表示 JBoss Web 平台将会是完整的 JBoss 应用服务器的一个子集，他补充说它的功能会在第一次发布时基本确定，其后不会变化太大。“如果你想要更多的功能，你可以上升到一个新的级别，选择一个全面的应用服务器，”Little 说，“如果总想着在一个轻量级的容器中提供更多的东西，最终它就会成为一个重量级的容器。因此，我们必须保持精简的 Web 配置文件，这有很多优点。”

总的来说，Little 认为，各种 JBoss 产品将会提供给人们所需要的多项新功能，但这些都不是最重要的事情。“人们对我们的最大的要求就是进一步提高拿来即用的能力、可管理性、还有易用性，”Little 说，“这不是任何新的或是酷的功能可以比拟的。这是 2010 年的主题，甚至也会是以后很长一段时间的主题。在必要时我们也会对性能做出改进，但估计这要到 JBoss 更普遍一些，至少需要普通的开发者也能够理解和配置 JBoss 的时候。”

开源 ERP：稳定的活跃度可避免风险

选择到合适的 ERP 软件能够大幅提高你的生产力、降低成本、改善运营，并将客户满意度提升到一个新的台阶。在选择 ERP 的决策流程中会有许多考虑因素，其中最主要的就是功能与价格，不过在开源 ERP 上，还要将某些特有的因素也考虑在内：

产品活跃度：开源 ERP 的主要特点就是产品的高透明度，可以免费下载试用。一款活跃的开源 ERP 产品代表它有较强的市场生命力。你可以通过 ERP 社区来了解某款产品的下载或使用人数，最新版本的发布日期，选择在一段时间内有稳定的活跃度的产品能回避掉一定的风险。



价格结构：不少开源软件都是可以免费下载、安装并使用的，但由于 ERP 系统的复杂性，因此还是需要有一些专业的技术支持。你不应去选择一款完全没有技术支持的 ERP 软件。虽然开源 ERP 系统的最大优势是无需支付许可证费用，但部分 ERP 厂商会以按月或按年收费的方式来提供后续支持服务。在选择之前，一定要清楚了解你需要支付多少费用，并可以得到哪些支持或附加值。

伙伴生态环境：如同所有权 ERP 实施，通常是由一名合作伙伴负责技术上的安装调试，让你的 ERP 系统开始运作，而 ERP 厂商自己很少直接介入。同样，开源 ERP 厂商在拓展全球合作伙伴网络上也投入了巨大的精力和资金。确定你所选择的开源产品，有至少 1 个以上的合作伙伴在你所处的地理位置，你需要与他们建立良好的关系，让他们协助你将 ERP 的投资回报最大化。

客户成功案例：相信每家公司都喜欢选择有成功案例的产品与厂商，因为这能为自己的项目实

施增加信心。在开源 ERP 项目上，你可以咨询厂商或合作伙伴，与他们的以往客户进行交流，清楚了解该软件或合作伙伴的强项与劣势，做出更有把握的选择。

清晰的开发蓝图：任何 ERP 决策在执行后都会对你的企业产生连续的影响，少则几年，多则十几年。因此，你所选择的开源 ERP 厂商必须要有清晰的产品开发蓝图，在未来一定时间框架内，有后续的升级或新版本开发路标。这一产品开发路标最好能公开在该厂商的官方主页上，并开放给开源 ERP 社区共同讨论交流，而不是隐蔽或晦涩不明的。

Sun 被收购 Linux 和 Solaris 将何去何从

甲骨文收购 SUN 后，由 SUN 引领的 UNIX 技术会何去何从也引发了业界的担忧。不过随着这项并购交易的获批，甲骨文高管也在上周三的新闻发布会上证实，他们将努力保持原来的产品线，Solaris 操作系统的用户也就不必为此太过担心了。

在 Solaris 操作系统方面，甲骨文公司目前已经是 Linux 操作系统的有力支持者。甲骨文拥有自己的企业版 Linux 产品（以红帽的 Linux 企业版为基础）。对于甲骨文公司首席执行官拉里.埃利森来说，Linux 和 Solaris 的相互排斥显然是一个错误的选择。

埃利森在一次网上直播的问答环节对这项技术发表了自己的看法"我不认为这是一个必须二选其一的命题，因为 UNIX 操作系统在高端领域就做的很出色。我想 Solaris 作为单机版计算机操作系统运行的时候会有所减少。Solaris 操作系统将更多的用在计算机集群上"。

在 Sun 的时代，OpenSolaris 的开放源代码可以用于独立的计算机。作为未来技术的实验平台，OpenSolaris 也代表了新一代的 Solaris 系统。虽然埃利森没有在他的讲话中具体提到，但他向大家介绍了 Linux 的未来规划。埃利森表示"我喜欢 Linux 系统，甲骨文一直是 Linux 的大力拥趸者，而 Solaris 又是一款历史更长，功能更强的操作系统"。

虽然埃里森预计 Solaris 会主要用于高端市场，不过按照它的发展规划也会向桌面系统延伸。同时埃里森也强调，高端市场是 Solaris 系统的主攻领域，可能是云或者 x86 或者 SunSPARC 服务器埃利森表示"我们认为 Linux 还需要很长一段时间才能迎头赶上。但是我们仍然热爱 Linux-我是一个 Linux 的爱好者，如果你需要 Linux 系统，我们将奉献给你世界上最好的 Linux 操作系统。如果你想要 Unix，我们也拥有世界上最好的 Unix 系统。再者这是两款不同的系统，我认为定位高端市场根本不存在任何问题"。埃利森对 Linux 与 Unix 的关系的评价与竞争对手-Unix 厂商惠普公司（纽约证券交易所代码：HPQ）的看法类似。除了 HP-UX，惠普还同时向用户提供 Linux 和 Unix 系统，他们将 Unix 系统专门定位为高端关键任务部署。

"我们从事开源业务已经很长一段时间了。我们一直是 Apache 的主要赞助者，并且也拥有自己的 Linux 操作系统—Unbreakable Linux。"埃利森同时也表示，同时拥有 Linux 和 Solaris 对于我们来说没有问题，我们希望能让两款系统都发展的更好。

社区扫描

****

微软认为 OpenOffice.org 是一个威胁

在一般人心目中，开源办公软件 OpenOffice.org 距离真正挑战微软 Office 还很远，算不上是一个强大的竞争对手。但微软似乎并不这么看。微软官网有一则招聘广告，软件巨人在为营销部门寻找“Linux 和 Open Office 竞争主管”。

在职位描述中，微软将 Linux 和 Open Office 视作是头脑中的“最重要问题（top of mind）”，并且要与开源社区和开源组织交战，从 Linux 和 Open Office 手中抢占更多市场份额。

看来，微软最近对开源爆发出的热情并不是真正认识到开源的价值，而是想获得内部知识，武装自己，再展开新的对抗。

Linux 基金会 CTO 跳槽 Google

著名的 Linux 专家 Theodore T'so 1 月 12 日在自己的博客中透露，他已经为 Google 工作数周，并对 Google 最近的举动表示赞赏和自豪。在这篇博客的评论回复中，他表示自己在 Google 将主

要从事内核、文件系统和存储技术的开发。当然，首要任务是 ext4 文件系统。Google 在经过大量基准测试之后，正在从 ext2 转移到 ext4。

T'so 是 Linux 社区的重量级人物。作为在北美的第一位 Linux 内核开发者，他组织了一年一度的 Linux 内核开发者峰会，并曾是自由标准组织的创始成员和主席。他也是 IETF 安全领域的重要成员。

技术方面，他毕业于 MIT，曾担任网络身份验证协议 Kerberos 的团队负责人。他也是 ext2，ext3 和 ext4 文件系统的核心维护者之一。此前，他供职于 IBM 实时 Linux 开发组，2008 年开始担任 Linux 基金会（Linux 的重要支持和标准化组织）的 CTO 两年。目前该职位正在空缺。

T'so 是继 Andrew Merton、Robert Love 之后又一位加盟 Google 的重量级 Linux 内核开发人员。

Facebook 成为 Apache 软件基金会黄金赞助商

Apache 软件基金会(ASF)迎来了最新的黄金赞助商，它就是著名网络社交平台 Facebook。据报道 Facebook 需每年捐助 4 万美金给这个开源组织。ASF 主席 Jim Jagielski 说：“赞助 ASF，将帮助我们现有项目的发展，孵化新的项目，促进社区发展和基础建设。”根据 Facebook 的 David Recordon，公司的技术平台都使用了开源技术建设，此举是为了回报社区。Facebook 已经是 ASF 的主动贡献者，为多个项目工作，包括 Thrift，Cassandra 以及 Apache Hadoop 的子项目 Hive。

其它的 ASF 赞助商包括白金赞助商 Google，Yahoo 和 Microsoft，他们需要每年捐助 10 万美金，HP(黄金赞助商)。

Firefox 的未来系于 Google 之手

根据 Net Applications 的统计数据，Firefox 现在已占据全球浏览器市场份额的 25%。Firefox 的成功迫使竞争对手改进产品，提高竞争力。过去两年中，微软、苹果和 Opera 产品性能都有引人注目的表现。

Google 现在是 Firefox 的默认主页和默认搜索引擎，Google 和 Mozilla 之间的交易为 Mozilla 带来了大笔收入，仅 2007 年就达到了 6600 万美元，占基金会总收入的 88%。这项交易将于 2011 年终止。现在 Google 已成为浏览器市场上的一位赛跑者，Mozilla 过于依赖 Google 的做法是否明智?

Mozilla 的副总裁 Mike Shaver 说，他看不出有任何理由让 Google 在 2011 年后终于与 Mozilla 的合作，称除了 Google 之外，Mozilla 还与 eBay、Amazon、Yahoo 和 Canonical 有合作关系。但 Shaver 也指出，Google 的继续合作关键在于 Firefox 保持现有的市场份额。

75%的 Linux 代码经由付费开发者产生

Linux 世界一直持有自由的崇高理想，但现实生活总是那么残酷：没有钱是万万不能的。APC Magazine 调查发现，Linux 的绝大多数内核代码是由大公司发展的，数字达到了惊人的 75%，真正的无偿志愿者贡献的代码只占大约不到 20%，绝大多数代码来自拿工资的公司员工，Linux 内核每天以 7000 行的数据递增，仅仅从 2.6.28 到 2.6.32 版本，变化就多达 55000 个，涉及代码 2.8 万行。

红帽贡献了 Linux 最多的代码，大约 12%，Intel 为 8%，IBM 和 Novell 提供 6%，甲骨文为 3%，虽然这些公司之间有着明显的竞争性，但对于内核的协作却一直很顺利。

红帽启动 opensource.com 社区站点

红帽公司最近启动了 opensource.com 这个非常酷的国际域名。这是一个由红帽主办的开源社区，网站使用了非常流行的开源内容管理系统 Drupal 搭建。据红帽公司 CEO 兼总裁 Jim Whitehurst 说：“这个并非红帽公司的网站，他是一个关于未来的开源社区网站”。他指出：“这是红帽公司回馈社区的其中一种方式”。

目前这个网站提供了 5 个频道，分别为商业、教育、政府、法律和生活。在那些频道下有一些关于开源的报道和文章。红帽公司 CEO 兼总裁 Jim Whitehurst 说，这只是最初的格式，将来会邀请一些社区开发者访谈。目前这个网站的内容红帽希望基于 CC 发布。

Ubuntu 10.04 默认搜索引擎更改为雅虎

Ubuntu 的主要支持机构 Canonical 今天宣布，该机构已经与雅虎达成共赢协议，作为协议的一部分，在 Ubuntu 中预装的 Firefox 浏览器中，默认搜索引擎将更改为雅虎。

Canonical 桌面团队主管 Rick Spencer 今天宣布了这一决定，只是协议的具体条目并没有公开。根据 Spencer 的说法，新的默认搜索引擎将“尽可能快地”出现在 Ubuntu 的开发版本中，并会在今年四月发布的 Ubuntu10.04 中正式启用。

Canonical 还没有宣布对已经发布的稳定版本会采取什么措施，现有的 Ubuntu 用户仍然可以保留默认搜索引擎？还是 Ubuntu 会提供相关措施更改现有的默认搜索引擎？这些目前还不得知。唯一清楚的是，升级至或下载安装 Ubuntu10.04 的用户都会发现其浏览器默认搜索引擎为雅虎。

届时，Firefox 工具栏和默认启动页面都会由 Google 更改为雅虎。当然了，用户仍然可以更改默认搜索引擎，为了方便用户，Canonical 还专门修改了 Firefox，用户只要修改默认搜索供应商，浏览器的起始页就会自动转换。

丹麦采用 ODF 为国家标准文档格式

经过 4 年的审议，丹麦国会最终通过了开放文档格式（ODF）为他们国家的标准文档格式。从今年的 4 月开始生效。这意味着微软的 Office Open XML 文档格式(OOXML)将失去更多的欧盟国家，还有像比利时，法国，瑞典，立陶宛，这些国家都采用 ODF 为他们的国家标准文档格式。尽管如此，OpenOffice 并不是唯一提供 ODF 格式的软件，微软的 office 2007 SP2 也将提供 ODF 格式支持。

而在中国，推出了 UOF，即标文通，所谓适合中国国情。但并没有广泛使用，如果与国外进行交换，格式还要进行转换。但我们看到，不管是什么行业，微软的文档格式还是一统天下。在未来的 3-5 年，恐怕还很难撼动微软的地位。

著名的内容管理系统 Alfresco 放弃 GPL

Alfresco 宣布更改他们的社区版内容管理系统许可协议，正式放弃 GPL，转向 LGPL 许可协议。这个更改是由公司的 CEO John Newton 在他的博客一篇文章上宣布的。他解释，公司使用 GPL 有三年多了，这是一个广泛使用的开源许可证，对于我们的项目发展起到了很好的作用。

Newton 称，公司品牌已建立，他们准备选择更加宽容的许可协议。对于 LGPL 和 GPL 之间最大的区别在于，LGPL 允许开发者以连接 LGPL 软件方式闭源。公司将 GPL 授权的软件卖给用户或集成商，而这些用户或集成需要集成我们的软件或发布品再销售，必须以 GPL 发布，这影响了用户的选择权。公司对此无能为力，因此选择 LGPL，可以有效解决这个问题。

Newton 称公司选择 LGPL 将不会影响专利软件所有人连接 Alfresco 软件库文件。他也引用了 JBoss 的推理为什么我们使用 LGPL。

五种开源协议的比较(BSD，Apache，GPL，LGPL，MIT)

当 Adobe、Microsoft、Sun 等一系列巨头开始表现出对”开源”的青睐时，”开源”的时代即将到来!

现今存在的开源协议很多，而经过 Open Source Initiative 组织通过批准的开源协议目前有 58 种(http：//www.opensource.org/licenses /alphabetical)。我们在常见的开源协议如 BSD，GPL，LGPL，MIT 等都是 OSI 批准的协议。如果要开源自己的代码，最好也是选择这些被批准的开源协议。

这里我们来看四种最常用的开源协议及它们的适用范围，供那些准备开源或者使用开源产品的开发人员/厂家参考。

BSD 开源协议(original BSD license、FreeBSD license、Original BSD license)

BSD 开源协议是一个给于使用者很大自由的协议。基本上使用者可以”为所欲为”，可以自由的使用，修改源代码，也可以将修改后的代码作为开源或者专有软件再发布。

但”为所欲为”的前提当你发布使用了 BSD 协议的代码，或则以 BSD 协议代码为基础做二次开发自己的产品时，需要满足三个条件：

如果再发布的产品中包含源代码，则在源代码中必须带有原来代码中的 BSD 协议。如果再发布的只是二进制类库/软件，则需要在类库/软件的文档和版权声明中包含原来代码中的 BSD 协议。

不可以用开源代码的作者/机构名字和原来产品的名字做市场推广。

BSD 代码鼓励代码共享，但需要尊重代码作者的著作权。BSD 由于允许使用者修改和重新发布代码，也允许使用或在 BSD 代码上开发商业软件发布和销售，因此是对商业集成很友好的协议。而很多的公司企业在选用开源产品的时候都首选 BSD 协议，因为可以完全控制这些第三方的代码，在必要的时候可以修改或者二次开发。

Apache Licence 2.0(Apache License， Version 2.0、Apache License， Version 1.1、Apache License， Version 1.0)

Apache Licence 是著名的非盈利开源组织 Apache 采用的协议。该协议和 BSD 类似，同样鼓励代码共享和尊重原作者的著作权，同样允许代码修改，再发布(作为开源或商业软件)。需要满足的条件也和 BSD 类似：

需要给代码的用户一份 Apache Licence，如果你修改了代码，需要在被修改的文件中说明。在延伸的代码中(修改和有源代码衍生的代码中)需要带有原来代码中的协议，商标，专利声明和其他原来作者规定需要包含的说明。

如果再发布的产品中包含一个 Notice 文件，则在 Notice 文件中需要带有 Apache Licence。你可以在 Notice 中增加自己的许可，但不可以表现为对 Apache Licence 构成更改。

Apache Licence 也是对商业应用友好的许可。使用者也可以在需要的时候修改代码来满足需要并作为开源或商业产品发布/销售。

GPL(GNU General Public License)

我们很熟悉的 Linux 就是采用了 GPL。GPL 协议和 BSD，Apache Licence 等鼓励代码重用的许可很不一样。GPL 的出发点是代码的开源/免费使用和引用/修改/衍生代码的开源/免费使用，但不允许修改后和衍生的代码做为闭源的商业软件发布和销售。这也就是为什么我们能用免费的各种 linux，包括商业公司的 linux 和 linux 上各种各样的由个人，组织，以及商业软件公司开发的免费软件了。

GPL 协议的主要内容是只要在一个软件中使用(”使用”指类库引用，修改后的代码或者衍生代码)GPL 协议的产品，则该软件产品必须也采用 GPL 协议，既必须也是开源和免费。这就是所谓的”传染性”。GPL 协议的产品作为一个单独的产品使用没有任何问题，还可以享受免费的优势。

由于 GPL 严格要求使用了 GPL 类库的软件产品必须使用 GPL 协议，对于使用 GPL 协议的开源代码，商业软件或者对代码有保密要求的部门就不适合集成/采用作为类库和二次开发的基础。

其它细节如再发布的时候需要伴随 GPL 协议等和 BSD/Apache 等类似。

LGPL(GNU Lesser General Public License)

LGPL 是 GPL 的一个为主要为类库使用设计的开源协议。和 GPL 要求任何使用/修改/衍生之 GPL 类库的的软件必须采用 GPL 协议不同。LGPL 允许商业软件通过类库引用(link)方式使用 LGPL 类库而不需要开源商业软件的代码。这使得采用 LGPL 协议的开源代码可以被商业软件作为类库引用并发布和销售。

但是如果修改 LGPL 协议的代码或者衍生，则所有修改的代码，涉及修改部分的额外代码和衍生的代码都必须采用 LGPL 协议。因此 LGPL 协议的开源代码很适合作为第三方类库被商业软件引用，但不适合希望以 LGPL 协议代码为基础，通过修改和衍生的方式做二次开发的商业软件采用。

GPL/LGPL 都保障原作者的知识产权，避免有人利用开源代码复制并开发类似的产品

MIT(MIT)

MIT 是和 BSD 一样宽范的许可协议，作者只想保留版权，而无任何其他了限制.也就是说，你必须在你的发行版里包含原许可协议的声明，无论你是以二进制发布的还是以源代码发布的。

行业观察

****

Android 撬动通信市场的新支点

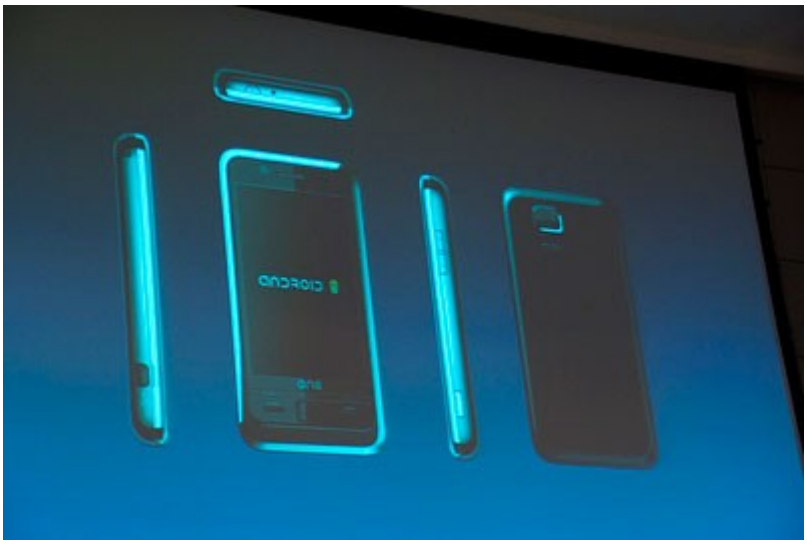
Android 就像是撬动整个通信市场的一个支点，支出了通信行业的新市场，使得通信业感受到了由 Android 带来的一股新技术暖流。

在 2009 年金融危机的“经济寒流”中，Android 就像是撬动整个通信市场的一个支点，支出了通信行业的新市场，使得通信业感受到了由 Android 带来的一股新技术暖流。

Android 何以成为一个支点

Android 能成为通信市场的一个支点，与其自身的特点优势有很大关联。

Android 最初是由 Google 开发的基于 Linux 平台的开源手机操作系统。它包括操作系统、用户界面和应用程序——移动电话工作所需的全部软件，而且不存在任何以往羁绊移动产业创新的专有权障碍，并迅速发展成为智能移动终端的操作系统。



开源(OpenSource)的特点在于改变以往由少数软件大厂垄断系统软件平台的现况，让众多内容开发商和开放软件供货商来分享共同利益，广泛增进客户使用经验。Android 有别于以往的手机及移动终端操作系统，其独具的开源性、系统廉价性和提供给第三方大自由度的创新空间，以及不受硬件约束的优势，获得了广大开放社群的力量。

Android 撬动了哪些市场

首先，随着厂商对 Android 认识的深入，生产的 Android 手机也愈来愈多。最早进入 Android 市场的台湾宏达电陆续在一年内推出了 G1、Magic、Hero、Tattoo 手机，三星也在近期推出 Galaxyi7500，摩托罗拉在早前推出新款 Android 手机 Droid，中国移动也以 Android 为基础开发了 OPhone 平台。这些都证明 Android 已经成为了智能手机市场的重要发展趋势。

其次，Google 推出的新一代网上平台 Android Market，让应用程序开发者轻松地进入这个“市场”，手机用户可在该平台寻找、购买、下载使用 Android 操作系统的手机应用程序和其他内容，也可在 AndroidMarket 上销售自己的软件，并通过交易获得利润。通过点击，就能轻轻松松赚钱。

再次，正是由于 Android 的系统开放性和服务免费，企业可免费获得源码并在此基础上开发新功能，延伸服务范围，加快研发速度，继而有效地节约成本，争取效益最大化。

谁在撬动 Android

一项新技术，虽然具有众多的先天优势，但如果没有一个助推力，那么它的成效也不显著。对于 Android 来说，它的助推器有社会组织，各个企业，还有行业主管部门。

Android 广大爱好者成立 Android 论坛交流讨论 Android 最新资讯，通过上传和下载来完善应用程序。为了推动 Market 和 Android 社区的发展，Google 悬赏 1000 万美元奖金举办“Android 开发者大赛”，来吸引更多人士参与其中。

各企业已意识到 Android 市场的巨大潜力，纷纷投入其中。中国移动的 OPhone 操作平台，就是基于 Linux 内核、采用的 Android 源码的移动操作系统。

台湾经济主管部门通过一系列方式来支持 Android 在台湾的发展。2008 年，台湾工业技术研究院进行了 Android 底层模块的研究；同时，台湾资策会也进行了 Android 上层应用的研究。

大陆方面，在工业和信息化部的支持下，海峡两岸 Android 技术及产业合作发展研讨会将于 2010 年 1 月 15 日-16 日举办，由中国电子信息产业发展研究院和台湾工业技术研究院联合主办。大会将邀请主管部门领导、电信运营厂商、海峡两岸技术专家和台湾行业主管部门等出席，就 Android 技术的开发应用、人才培养和对产业发展影响等话题展开研讨。

2009 是 Ubuntu 年

作为 ZDNet 博客 Linux and Open Source（Linux 和开源）的专栏作家，开源专家达纳·布兰肯霍恩(Dana Blankenhorn)近日撰文写道：

2009 年，如果说有一个词比微软更加让开源读者关注的话，那么一定是 Ubuntu。Ubuntu 并不是最盈利的 Linux 系统，而且也不是最为流行的 Linux 发行版。但由于 Canonical 商业公司，Canonical 公司有影响力的总裁马克·沙特尔沃思（Mark Shuttleworth），以及他们在桌面系统市场的雄心，Ubuntu 系统就成为了开源读者最为关心的新闻词汇。

今年初，我只是模模糊糊地意识到这一点。不过，我逐渐地坚定了这种看法。Ubuntu9.04 将于星期四提供下载——这是我坚定这种看法的原因之一。保罗·鲁尼（Paula Rooney）的新闻在 Ubuntu9.04 正式到来的前一天发布，这则新闻业也成为 ZDNet 今年的第 12 大热点新闻。

2009 年 4 月 23 日，Canonical 公司宣布，自美国东部时间 12 点，用户可以自由下载 Ubuntu 桌面版本 9.04。Canonical 公司还宣布同时发布 Ubuntu 服务器版本 9.04 和 Ubuntu 上网本特别定制版 9.04。

Ubuntu 仍将扮演次要角色吗？——这是今年 7 月份我来到中国·台湾发表的文章。我发现，这里是 Windows 系统的世界，Ubuntu 用户寥寥无几。这则文章的回复数达到了 386 个，它也因此成为 ZDNet 今年的第 5 大热点新闻。对于这个问题，有些人认为这是短期的，有些人则认为这是长期的。但我最喜欢回答是：Linux 将永远是操作系统的未来。

Ubuntu Karmic Koala（Ubuntu 9.10）正式发布——我在鲁尼之前发布了这则新闻，并且得到了 174 条回复，以及 54 个投票。这则新闻业成为 ZDNet 今年的第 3 大热点新闻。

2009 年 10 月 30 日，Canonical 公司宣布，代号 Karmic Koala 的 Ubuntu 9.10 操作系统式发布并上线。

下个十年开源操作系统将如何发展

时光飞逝，21 世纪的头一个十年就这样过去了。而开源软件也从最初的地下运作发展成了如今的主流商业模式。虽然商业软件还在后面苦苦追赶，但是很显然，它的经营模式已经跟不上时代的发展了。

从 2010 年开始的未来十年将是开源操作系统的青年阶段，而开源操作系统在这个阶段如何发展，将对它的未来至关重要。

开源软件被定义为描述其源码可以被公众使用的软件,并且此软件的使用,修改和分发也不受许可证的限制。开放源码软件通常是有版权的，它的许可证可能包含这样一些限制：著意的保护它的开放源码状态，著者身份的公告，或者开发的控制。“开放源码”正在被公众利益软件组织注册为认证标记，这也是创立正式的开放源码定义的一种手段。当前越来越多的企业选择了开源软件。关于开源软件的未来发展，现在业界的主流观点是开源软件将会被混合化。大部分商业软件将会采用开源的方式。所有的软件表面上看起来都是开源的，他们使用一种复合许可证，你可以像开源软件一样使用它，但是要是要为底层开发者支付一些费用。这看起来比较可行，因为可以为开源软件建立起合理的商业模式。而这也在云计算发面得以体现。将硬件与开源操作系统捆绑在一起。从用户的角度来讲，开源与否对于他们不重要，在使用上 Google Nexus One 和 iPhone 有区别吗？用户购买了一段时间的云计算服务使用权，他不会关心这些钱在软件提供商和服务提供商之间是如何分配的。而实际上，开源操作系统的传统领域造成了很大的冲击。商业软件有一定的生命周期，你必须要为使用下一代产品重新付钱，而开源软件可以使用户避免陷入付费升级的怪圈。互联网是开源软件的发源地，而随着新一代互联网的发展，尤其是 Google“一切皆通过网络”概念的提出，开源软件必将迎来新一轮的发展高潮。

提起开源软件，我们就不得不提到开源软件的最大敌人---微软。而最近一年，微软对于开源社区的态度有了很大的转变。微软到底是开源的朋友还是敌人？如果从微软近几年的动作来看，真的无法说清。一方面，微软拼命向开源领地摇动橄榄枝，捐赠代码、支持像 Apache Software Foundation 之类的开源企业，就好像它跟开源天生一对似的。事实上，早在 2006 年 11 月，微软

就和 Novell 签署了互操作性协议——值得说明的是，2 年过去了，这件事情引起的争议和抵制依然强烈。之后，微软和越来越多的开源厂商签署了互操作协议。而就在一个月前，Linux 领域的重量级厂商红帽(Red Hat)也与微软签署了合作协议，加强虚拟化平台互操作；另一方面，它仍然对它那些基于开源技术的专利实行收费，而且它的专利许可的运营策略与开源的运营有着天壤之别，况且微软还动辄就跟开源对簿公堂，微软曾声称开源软件侵犯了超过 200 个微软专利。那是在 2007 年 5 月，随即这一说法在开源界引起了轩然大波，包括 Linux 创始人、微软的合作伙伴 Novell 等在内的诸多开源人士都提出了激烈的批评，这场争论直到 2007 年底仍未彻底平息。在过去的一年，微软首先根据 Linux 采用的 GPL 发布了三个驱动程序，意味着将 2 万行装置驱动程序代码贡献给了开源社区。在年底迫于压力又将其 Windows7 下载工具开源。关于微软更多的开源动作，请参看：微软称开源使其比任何时候都忙。然而无论微软采取怎样的开源策略，无可否认的是开源公司、开源开发者及其拥护者已经发展得越来越强大，已经拥有了很成熟的商业模式。尤其是 Linux，它驱动了世界上许多大的网络比如 Google。而且越来越多的开发者和公司在选择使用开源的工具和软件。

IBM 大型机：究竟是过时还是涅槃重生？

回顾 IT 行业的热点新闻，会发现某些大型企业选择运行 UNIX,Linux 甚至 Windows 操作系统的服务器来替代 IBM 大型机的消息。

这提醒我们小规模的服务器-甚至是 Windows 服务器所能承担的工作负载已经远远超过了我们过去的想象，而在价位上与大型机差别很大。UNIX，Linux 和 Windows 服务器满怀信心的渗透到大型机驰骋的领域已经有相当一段长的时间了。

这并不是说大型机的日子已经屈指可数-而是说明在经济衰退期间很多企业选择相对低廉的商用服务器作为替代，IBM 的 System z 大型机系统的销售因此受到严重的影响。

但还有一些令人些奇怪的消息：就是在圣诞节前 IBM 公司宣布韩国最大的信用卡公司-BC Card 决定使用 IBM System z 大型机来支持其信用卡支付系统，而不是选择惠普和甲骨文的产品来进行替代。BC Card 公司的首席信息官 Jeongkyu Lee 在回答记者提问时这样说道"我们之所以选择 System z 大型机是因为通过 IBM 的大型机软件解决方案可以保证服务质量和业务运营的持续性，并且能在今后几年内为公司带来经济回报"。

笔者认为企业用户青睐 IBM 大型机运行持续性的优势也在意料之中。这也正是大型机的魅力所在-"零宕机"就是 System z 大型机中"Z"这个字母代表的含义。

更奇怪的是有人认为在未来数年内随着经济的复苏，IBM 解决方案将被认为是一个低成本的选择。特别是当你考虑用什么新的 IBM 系统来进行替代时。根据 The Register 网站的说法"BC Card 目前青睐 System z 大型机解决方案，因此他们大幅倾销惠普和 Sun 制造的 Unix 服务器"。

IBM 时不时会因为某些事而遭遇诉讼，但低成本通常不是被起诉的理由之一。去年在 IBM 的 System z 大型机季度销售额遭遇 39%的灾难性下滑后，IBM 在 8 月份宣布 IBM 解决方案版本战略。该解决方案版本推出的目的是让 System z 大型机的价格不再像以前那么高昂，否则企业用户就会去选择使用 UNIX 或 Linux 服务器来替代 IBM 大型机。IBM 公司对这一战略进行了积极的推广并宣称"IBM 大型机卓越的性能和灵活性将更容易得到企业用户的认可和接受"。

与 BC Card 达成的交易就是以 System z 解决方案版本为基础的。这个解决方案包括 IBM 中间件（包括 DB2 数据库），CICS，WebSphere，信息管理和 Tivoli 软件。IBM 技术支持服务也包括在内。根据 Register 的说法，这意味着打包购买的价格比单独购买的价格低了 50%到 80%，这样价

格也就比运行 HP-UX 操作系统的惠普动能服务器高出不到 20%。

IBM 公司表示这是他们近 10 年来首次从 UNIX 到大型机的过渡，因此解决方案版本战略似乎很奏效。降低价格最终能吸引用户。IBM 还没有完全采用过去电视广告的促销方式来推动 System z 大型机销售额的增长，但很可能在接下来的营销计划中会加以实施。

只有当 IBM 对 System z 大型机的销售不再恐慌并恢复大型机产品的售价，惠普，甲骨文和其他生产 Linux 和 UNIX 服务器的厂商才能放下心来。

2009 年开源软件发展遍地开花

开源软件在 2009 年逆势飞扬，在多个领域取得了成绩，Xen、Ubuntu 和 Android 等开源产品大放异彩，云计算、SaaS 的商业模式也带动了产业界的发展。

一、经济危机使性价比高的开源软件获得了更多的市场机会

经济危机使开源软件获得了更多的市场机会，由于对采购预算的缩减，越来越多的企业开始重视使用性价比高的开源软件。

在美国，对开源的支持也使奥巴马被外界冠以“开源总统”之称，他上台伊始就委托 Scott McNealy 等人为美国新政府起草“开源白皮书”，把传播开源技术作为对处于金融危机和经济衰退漩涡中的美国的解困对策之一。同时，开源软件的安全性也得到了正确的认识，开源白皮书指出：“从操作系统到中间件，到数据库，到浏览器以及 Java 等，比之私有软件产品，开源软件产品在其生命周期里面临更少的安全问题”。美国白宫网站 Whitehouse.gov 的内容管理系统(CMS)现在已换成了开源的 Drupal 软件。就连一直固守专有软件的美国国防部现在也开始向开源软件敞开大门。

二、云计算、SaaS 潮流下，开源软件的服务特性更易于被推广

“开源软件和云计算软件像孪生兄弟一样的不分家”，Red Hat 总裁兼 CEO 卫赫士(Jim Whitehurst)在不同场合多次这样说。在今年的“开源·云计算”大会上，卫赫士先生访华并发表演讲，表示目前全世界有 90%以上的云计算部署在开源平台上，可以认为云计算的特性决定了开源计算在云时代的领导位置。的确，伴随着“云计算”在今年的不断升温，很多中小企业开始使用云计算，利用网络托管的服务，减少了在技术基础架构上的大量前期投入，仅需付出按需使用的“水电费”。

开源软件具备丰富的资源，可以整合出完美的应用，开源软件可以以提供服务作为其赢利模式，基于服务模式可以全面展现开源软件的精髓和商业模式，Google 公司是其中一个成功范例。SaaS、SOA、云计算等都是非常大的系统和架构，全部自己开发并不现实，基于开源软件可以极大节约实现的时间和投入；如果全部采用商业软件，将面临高昂的许可费用，企业的初期投入或总体投入会非常庞大，特别是初创互联网企业，应更多采用开源软件；采用商业软件，从长远来看，没有可能掌握核心技术，采用开源软件，可以帮助服务商更快掌握和部署整体技术架构，站在巨人肩上实现更快发展，也有利于更快掌握核心技术。

在 SaaS 的概念逐步得到认可的形势下，我们需要进一步认识和实践软件的服务本质，基于开源软件形成整体有效的服务体系。

三、开源软件使芯片产业实现软硬件协同发展

软硬件协同发展的趋势在今年日益明朗，芯片巨头英特尔一方面在硬件上继续沿袭“摩尔定律”加快芯片的发展，另一方面也重视起软件，在 IDF 开发者论坛上推出 Moblin 2.1 开源操作系统的

测试版。英特尔以前是在配合微软做软硬件协同，两大产业大亨将 Wintel 联盟的优势发挥得淋漓尽致，到了现在英特尔充分认识到属于自己的自主、可控的操作系统的重要性。也只有基于开源的优势，硬件巨头英特尔才能较快地拓展软实力，开发出自己的操作系统。

Moblin 2.1 将面向智能手机、上网本、上网机(nettop)、MID(移动上网设备)以及车载系统等多种设备，在多个领域与微软展开竞争。

四、Linux 操作系统遍地开花

2009 年，Linux 的发展仍然是开源界的亮点，在这一年，桌面 Linux 的驱动、应用和兼容性方面的不足都得到了很大的改进。如上提到的 Intel 推 Moblin，而谷歌宣布推操作系统是开源界的重大新闻，它基于 Linux 内核的特质将极大促进 Linux 桌面的推广。

智能手机操纵系统方面，Google Android 手机操作系统的崛起也同样不容忽视。Gartner 在关于 2012 年全球智能手机操作系统市场格局的预测中，还提到“各类 Linux”智能手机将为 2800 万部，占 5.4%“，“各类 Linux”指的是 NTT Docomo、NEC、Panasonic、Access(Palm Source)开发的 Linux 系统，以及 Motorola 开发的不同于 Android(Droid)的 Linux 系统，Nokia 开发的不同于 Daemo 的 Linux 系统，这些开源的力量也在支持智能手机发展中的高速成长。

对甲骨文说不 美公司换装开源软件节约成本

要说当前的经济衰退有什么好处，那就是彻底揭露出各大软件公司采用的伎俩——通过多年许可协议，牢牢锁定客户。按照这些协议，每年的许可费都在上涨，几乎从来没有降过，哪怕客户缩小了业务规模，也是如此。

各大软件公司的策略主要采用折扣手段，导致“搁置软件”(shelfware)和续约陷阱，并辅以额外收费的手段。比如针对在企业部门之间迁移软件，使用最新硬件，或者客户通过互联网访问内部数据等情况额外收费。

这门“艺术”在过去四十年来不断得到完善，并且让许多人大发其财。今天这一幕仍在上演，即便是在一个不断改善、但仍深陷困境的市场。去年该情景至少上演两次：甲骨文和 SAP 针对牢牢锁定的客户群将价格提高整整 18%。

但是，许多精明的企业客户已经采用了新的软件商业模式，避免大型软件缺乏灵活性、成本高的弊端。

詹姆斯·西姆斯(James Sims)是总部设在加州莫德斯托、年收入达数十亿美元的食品售企业

Save Mart 的 IT 副总裁兼首席信息官。他已经受够大型软件公司一贯以来的手段。在考虑开源软件之前，Save Mart 为 250 家连锁店、3 个仓库和 2 万名员工及货运车队服务的 IT 系统被甲骨文、微软及其他软件厂商的大型软件许可证牢牢绑住了手脚。西姆斯表示，多年来，甲骨文对 Save Mart 公司的态度尤其差，总是不屑一顾、傲慢无礼的面孔。于是他告诉对方，Save Mart 会尽一切努力，不再与甲骨文打交道。结果，他说到做到了。

西姆斯把该公司 IT 基础架构上的专有软件统统换成开源技术，部署了 SuSE Linux 操作系统和 Ingres 开源数据库，还部署了 Hobbit 开源监测工具，用于监测 Save Mart 的服务器、应用软件和网络运行状况。他还与 Novell、红帽和 Ingres 等厂商建立联盟，支持使用开源软件。

据西姆斯声称，开源软件不但与专有软件一样好用，在许多情况下，甚至比专有软件还好用。他表示，自从用开源解决方案取代甲骨文、微软和惠普的专有软件后，每年为公司节省 50%以上的 IT 成本。

西姆斯说：“太多 IT 部门选择了现行的解决方案。唯有时间会证明我们另辟蹊径到底有没有错。很幸运，食品利润能达到 2%，所以成本相差 0.1 甚至 0.01 个百分点，也有很大的关系。我们的 IT 部门运营费用占收入的 0.37%，是食品零售商同行平均 IT 支出的一半。但是我们能够以低于其他食品零售商的成本，提供更高的质量、更好的服务和更多的选择。我们省下来的每一分钱都直接为客户降低了成本。”

丹尼尔·弗拉克斯(Daniel Flax)是总部设在纽约的投资银行 Cowen Group 的首席信息官，由于对成本高、缺乏灵活性的专有软件感到不满，他也开始考虑开源解决方案。他希望可以把这些成本高昂的解决方案换成更具有成本效益的开源解决方案，并且在不牺牲质量、性能或不用投资遗留系统的前提下，完成更换工作。

他的目的是通过一种新的方法来提高成本效益、增强竞争优势，并且为股东带来最大回报。在企业内部，他要求 Cowen 的每个 IT 项目至少包含以下三个要素中的两个——移动性、软件即服务(SaaS)或开源软件。

弗拉克斯认为 Ingres 数据库性能与专有数据库性能一样好，而成本只有其十分之一。他曾为一个重要的程序交易项目比较了众多数据库的综合成本、成熟性和功能后，还是选择了 Ingres。他也关注过 MySQL，却担心 Sun 被甲骨文收购后 MySQL 前景不明。

由于选择了商用开源解决方案，Cowen Group 没必要支付变更费(使用权由一方转到另一方时要付这笔费用)、多核费用、站点许可证续费或者节节攀升的支持费用。相反，该公司能够对预算的大部分进行调整，投入到满足 Cowen 和弗拉克斯目标的创新 IT 项目上。

总部设在澳大利亚的食用油和人造黄油公司 Peerless Foods 自行开发了企业资源规划(ERP)系统，连面向全自动化仓库的一套物理控制系统也是自行开发的。首席信息官亚德里安·汉密尔顿(Adrian Hamilton)失望地发现，市面上没有商用软件方案让 Peerless 可以管理从发票到产品分销的一切。他考虑过商业模块，但被高昂的成本吓到。

随着业务不断增长，Peerless Foods 需要一个更稳定、更灵活的平台来更新生产环境、支持将来拓展。尽管面临相似情形的公司往往直接选择专有软件厂商，但 Peerless 决定从头开始自行开发 ERP 和商业智能解决方案。

之所以后来选择开源技术，是因为它仅以六分之一的价格，就能提供最好的功能。Peerless 的开发人员借助 Ingres OpenROAD 应用开发环境构建自己的 ERP 系统。如今汉密尔顿估计，公司每年可以省下三、四十万美元的许可费。对于一家年收入 4 亿美元的企业来说，这相当于收入的 1%。

Peerless Foods 眼下甚至在考虑将其内部开发的 ERP 系统作为一款开源或商用现成产品来进行发布。

要说服有些主管接受非传统技术还是有些难度。比如，他们会觉得企业内部 IT 部门使用大型专有软件产品比较稳当，而没有意识到其实是为自己围起了樊篱，放弃了自由创新的机会，也无法切实节省资金。他们还没有充分认识到开源和 SaaS 解决方案已经变得多强大、多可靠。

像 Save Mart、Cowen Group 和 Peerless Foods 这些开路先锋为我们指明了企业中使用非传统 IT 技术的最佳途径。相信会有更多厂商会在未来对大型软件公司大声说不。开源技术已具备能力在 IT 领域塑造新的经济格局。

甲骨文 MySQL 和 Sun 服务器发展计划

甲骨文发布了对收购 Sun 后的产品承诺。甲骨文表示，将积极推进 MySQL 开源数据库的发展，而不是任其自生自灭。同时甲骨文宣布了 Sun 硬件和其它软件的发展计划。

据甲骨文公司首席社区架构师 Edward Screven 介绍，甲骨文还计划继续投资和维持 OpenOffice.org 的独立性，同时还将推出一个单独的类似于谷歌 Docs 的云生产力套件。目前，Sun 的 OpenOffice 一直是微软 Office 的有力挑战者。

此前，由于和甲骨文的商用数据库产品冲突，许多用户担心甲骨文将逐渐取消 MySQL。因为此前 MySQL 凭借开源的思想和理念在数据库领域谋得了一席之地，并在一定程度上构成了对甲骨文的威胁。将 MySQL 收至旗下后，用户最担心的是甲骨文将其“冷冻”起来。

并且，甲骨文一向喜欢融合被收购公司的产品和技术，此前就对 People Soft 的收购就是一个比较典型的例子，尽管甲骨文保留了 People Soft（仁科）的商标，但对这家公司知之甚少的企业很难再感受到仁科存在。这也造成了前期欧盟迟迟不通过甲骨文收购 Sun 交易审查的主要原因。

为此，甲骨文不得不做出有限的让步，2009 年 12 月，甲骨文做出了 10 项具体承诺以此来支持 MySQL 的后续发展。甲骨文表示，未来 3 年内至少投资 7200 万美元用于开发 MySQL，以及继续发布这款开源软件的最新技术研发成果。在未来 5 年内允许其它技术厂商继续在各自产品中授权使用 MySQL。

现在，欧盟通过并购审查后，甲骨文方面表示将继续 MySQL 的独立的销售队伍，同时提高

MySQL 的代码，支持和与其他 Oracle 应用程序的兼容性。Screven 将在开源软件部门负责 MySQL、OpenOffice.org 和其他开源开源应用。

在上周三举行的网络直播发布会上，Screven 和其他甲骨文的官员表示，甲骨文收购 Sun 的许多一流技术后，将使合并后的公司提供“完整、开放、集成”的系统。“甲骨文和 Sun 合并后对 IBM 是一个巨大的挑战。此外，合并后的甲骨文-Sun 还是企业开源技术的最大支持者。”Screven 这样评价道。甲骨文还在发布会上公布了其它一些软件及技术的发展规划。

Java：甲骨文计划“扩大和加强 Java 的范围”，同时实现 Java 平台的整合和简化。具体来说，就是推出针对桌面电脑的 Java 标准版客户端 7，同时推出移动版本的 Java ME，它与桌面版本相兼容，以减少程序员的工作。

甲骨文官员还表示要使得 Web 网站开发者方便地使用 JavaScript。这些举动将有助于壮大 Java 开发人员社区，目标是达到 1000 万人。

OpenOffice.org：Screven 表示，OpenOffice.org 将作为一个独立的业务部门来管理，Sun 的开发和支持团队会继续保留。甲骨文将继续支持免费的 OpenOffice.org 社区。但是甲骨文还计划推出名为 Oracle 云办公的软件套装，甲骨文已经开发这个软件有一段时间了。但是原来 OpenOffice.org 支持的 StarOffice 没有被提及，这是原来用于和 IBM 竞争的一款产品。

Solaris：这是 Sun 在 2005 年宣布开源的服务器操作系统。甲骨文表示计划增加在 Solaris 上的投资，目标是使其能够同时运行数千个 CPU 线程和处理数 TB 的内存。

Linux：甲骨文支持 Red Hat Linux，并已经为成千上万的用户部署了 Linux，甲骨文公司承诺将继续投资 Linux 和 Solaris。

至于甲骨文所短缺的服务器等硬件设备，甲骨文官员表示将继续投资于 Sun 的多线程 UltraSparc 处理器，并将其用于 Niagara 服务器，以及基于富士通开发的 Sparc64 处理器的 M 系列服务器家族。此外，甲骨文还将继续开发和销售 Sun 的 x64 服务器，它使用了来自英特尔和 AMD 的处理器。

甲骨文在将 Sun 的硬件收于旗下后，还会结合原先双方的优势，推出整合型的产品，比如数据库专用机。它将甲骨文的数据库和 Sun 的邮件结合起来，并进行了优化。IDC 的分析师为此表示，甲骨文对双方的优势进行整合后，能使得软硬件更好地工作，同时有利于尽快将产品推向市场。

由于甲骨文和 Sun 合并交易等待批准的过程中耽搁了一段时间，IBM、HP 等公司在这段时间内疯狂地拉拢原来 Sun 的客户转投阵营，因此甲骨文在宣布新的规划和承诺后，需要尽快将信心带给老客户，以阻止他们走向竞争对手。

下一步，甲骨文表示将会继续瞄准零售和电信等行业的客户，为他们提供整合软硬件一体的解决方案，同时继续出售通用型服务器。

本期推荐

****

谷歌 Android 被 Linux 内核除名

作者：袁萌

2 月 3 日， Linux 内核开发者 Greg Kroah-Hartman 将 Android 的驱动程序从 Linux 内核“状态树”（“ staging tree”）上除去，从此， Android 与 Linux 开发主流将分道扬镳。原因是，“ no one seemed to be working on it”，难怪 Linux 内核开发团队决定不要它了。呜呼！

后果如何？今后，在开源的路上，谷歌 Android 的硬件驱动从何而来？谷歌喜欢开源，是喜欢它自己所喜欢的那种开源，而不是 Linux 的开发主流。谷歌气大财粗，自己去单干吧！

国内开源界，不要跟在谷歌屁股后面瞎吹了。说到底， Linux 是草根事业，气大财粗也不灵。近日，谷歌 CEO 又口出狂言，要给中国政府“一点儿压力”，想改变谷歌目前的这种被动状况。我看，根本没门儿。

什么是 Win+nux 客户？所谓“ Win+nux” 是一个新杜撰出来的组合词，就是 Windows+Linux 的缩写组合，两者相互融合的生成物。此举，有何实际意义？

根据 www.w3counter.com 的最新统计数字，今年 1 月份，全球 Linux 用户群体突然间蒸发（缩小）了约 1/4 ，从占有率 2.14% 猛然下降到 1.63% ，这可不是一件小事情，人们从未见过这等怪事。在此期间， XP 下降了 0.53 个百分点， Vista 下降了 0.49 个百分点，唯独 Windows 7 增加了 2.31 个百分点，占有率升至 9.11% 。显然，在这期间，部分 Linux 用户转向了 Windows 7 。我们再看一串数字：2009 年 9 月， Linux 占有率为 1.84% ；2009 年 10 月， Linux 占有率为 2.14% ；2009 年 11 月， Linux 占有率为 1.84% ；2009 年 12 月， Linux 占有率为 2.14% 。

由此，我们不难看出，部分用户在 Windows 和 Linux 两者之间转来转去，并不固定。他们既是 Windows 用户，也是 Linux 用户，反之亦然，因此，形成了一个新的边缘群体，即 Win+nux 用户群。

实际上， Win+nux 用户多半具有多台计算机，配备了不同的操作系统。 Win+nux 用户并不特别偏向（喜好）哪一种操作系统，态度中立、务实，如同对待自己的左、右手一样。从全球范围来看， Win+nux 用户群体也许有数千万人之多。而且，我们可以这样设想，这种群体将会继续扩大和发展下去。发展 Linux ，硬是要与 Windows 对着干，缺乏事实根据。联想 Windows+Linux “ 二合一”计算机就顺应了这种发展潮流。喀纳斯（ kanas ）自由软件专卖店的 Live 自启动 U 盘的经营思路也是如此。有人说，发展 Linux 是为了给用户多一种选择。严格地讲，发展 Linux 是为了让用户在这两者之间自由地转来转去，无所谓哪一种操作系统更具优势，让他们在这两者之间，选择一种操作系统，实际上，这两者都不错。由此看来，在今 年一月份，我们不能说在全球范围内， Linux 吃了一次大败仗，几乎全军覆没。到今年 4 月份，当 Ubuntu 10.04(LTS) 发布之后，说不定广大的 Win+nux 用户又要回到 Linux 操作系统上呢！ Win+nux 用户不是两面人，他们都是很普通的自然人。 Windows 和 Linux 都是计算机操作系统，不应当是两家人。这就是应有的结论。

有人也许会说，你这是在抹稀泥。不是我在抹稀泥，而是客观事实就是这样的。人的大脑要反映客观实际，杜撰出这个新词 Win+nux 用户，反映客观实际，不是我的错。你说呢？

自由软件的核心

ChinaUnix 网友： Jean. Love

自由软件的核心，是源代码开放吗？是人人可以自由使用吗？是人人可以自由的修改再发布吗？是没有人可以占为己有用于商业盈利吗？这些自由软件的特证里面，哪个才是最根本的属性？

一种非常可能的答案是，以上都不是。最根本的属性，是保证"人人可以自由使用/自由的修改再发布"等许多基本软件自由的，公共许可协议：GPL。

GPL 是一个"法令"式的条文，它规定了看到了 GPL 许可证的人必须被认为已经读了 GPL 协议并且能自动的遵守它，不遵守的将在法理上处于不利地位。为什么要这样做？因为如果说商业软件剥夺了人们太多的资金和控制权，人们只能听凭商业软件的定价，那么自由软件能够还给软件用户所有的根本的自由。但是如果自由软件的意义只是如此，那么就不会和非自由的开源软件拉开距离了。

因为有这样一种风险，自由软件从诞生之日起，扮演了软件世界救世主的角色，它拯救了无数人使用软件的自由。但是，这个救世主可能被背叛和出卖。自由软件本身可以被用于商业目的，可以被用于垄断盈利，也可以借用自由的借口，再达到了商业目的以后，宣布今后的版本不再自由，而是推出一个封闭的商业版本。出卖救主的犹大无处不再，他们的借口种种，诱惑种种。于是，为了防止此类事情的蔓延和扩大，必须有一部自由软件的宪法，那么它就是 GPL，这个宪法的创始人就是 FSF 之父，理查德.斯托曼。

但是光有 GPL 好像还是不够的，因为软件专有的力量过于强大。就像美国的梦山都种子公司的转基因农作物专利大棒一样，它规定了即便是自然作物接受了专利作物的花粉，那么也算是自然作物的耕种者侵犯了该公司的专利。专利和专有制度是如此的强大和邪恶，并拥有传染性。那么在这种事情发生在自由软件的世界之前，我们的斯托曼圣人就预感到了这种危险的可能性，所以以进攻的姿态推出了让自由软件的"自由"不可能受到侵害的，传染性的公共许可协议：GPL3。不管有多少人反对这个病毒式的协议，斯托曼保证了即使留下一个自由的种子，它也可以通过无穷的病毒式的传播生存下去。为了和非自由对抗，不得不做出这样的选择。

而那些非 GPL 的所谓开源软件呢？他们或者是被商业公司用来扮演低端产品的角色，或者干脆就是在达到了一定的用户数量以后，被转型为封闭的商业软件，彻底的抛弃和背叛 Open Source 自身。正是由于 Open Source 本身并不能保证不损害 Open Source 本身，它的生命力就远远不如 GNU 软件。Open Source 运动的发起人 ESR 是一位著名的，斯托曼的背叛者。他开创的所谓灵活的开源世界本身是个不完全的商业软件世界，是非自由的，是可以损害开放型原则的，它依靠的是开发者本身的道德---显然这不可靠。而 GPL 尤其是 GPL3 则是战斗的，诅咒式的，具有强大的法力，可以保证道义上自由软件的精神永存并且永不被打败。ESR 认为开源软件应该是集贸市场式的，松散的联盟；而斯托曼则坚持彻底的自由，从操作系统到编译器到应用程序，并且应当是强制的。因为如果基础软件不是自由的，那么所谓的 Open Source 的根基其实就是不自由的，开放也是不牢靠的----Open Source 随时可以沦为专有软件的奴役对象。

斯托曼是寂寞的，因为他是 FSF/GPL 这座自由软件大教堂的建设者----菜市场模式不能保证自由不受侵害，自由软件模式的根基仍然是"大教堂"+各处自由的教会。没有这个大教堂对自由的保障，所有的 free 都是空谈。GPL3 对背叛者关上了大门，因为这个规定是如此的严格和充满诅咒，以至于制定者自身都完全不可以违背它。天使如果不比恶魔更加强有力，那么天使就不能战胜恶魔。如果自由软件的世界真的是伊甸园的话，那么斯托曼大牧师在园子里到处撒了毒蛇吃了就会归天的药丸，所以伊甸园里面没有了毒蛇。

技术新知

****

使用 port 基本安装 PostgreSQL 笔记

ChinaUnix 网友：zeisoctopus

1. 假设环境

pgsql.home.net 192.168.1.3 255.255.255.0

文中的 port 选项是本人喜好，不是预设值或建议值

2. 更新 FreeBSD 中的 port tree

$ su -
$ portsnap fetch update
$ exit

如果你从未用过 portsnap，改为

$ su -
$ portsnap fetch extract update
$ exit

3. 登入 pgsql.home.net 后从 port 安装 perl 5.10.1

系统默认用 perl 5.8.x，如果你想 postgres 用 perl 5.10.x，有两个方法
i) 编译 postgresql 之前先编译 perl 5.10
ii) 在 /etc/make.conf 加入一行 PERL_VERSION=5.10.1

我习惯先编译 perl 5.10

$ su -
$ cd /usr/ports/lang/perl5.10/
$ make config

```
|  |     [ ] PERL_MALLOC   Use Perl malloc                  |  |
|  |     [X] PERL_64BITINT  Use 64 bit integers (on i386)   |  |
|  |     [X] THREADS       Build threaded perl              |  |
|  |     [ ] MULTIPLICITY   Use multiplicity                |  |
|  |     [X] SUIDPERL      Build set-user-id suidperl binary |  |
|  |     [X] SITECUSTOMIZE  Run-time customization of @INC  |  |
|  |     [X] USE_PERL      Rewrite links in /usr/bin        |  |
```

```
$ make
$ make install
$ exit
```

4. 在 pgsql.home.net 从 port 安装 postgreSQL 8.4.1

```
$ su -
$ cd /usr/ports/database/postgres84-server
$ make config
```

```
┌──────────────────────────────────────────────┐
|          Options for postgresql-server 8.4.1 |
|  ┌──────────────────────────────────────────┐ |
|  | [X] NLS            Use internationalized messages |  |
|  | [ ] PAM            Build with PAM support (server only) |  |
|  | [ ] LDAP           Build with LDAP authentication support |  |
|  | [ ] MIT_KRB5       Build with MIT's kerberos support |  |
|  | [ ] HEIMDAL_KRB5   Builds with Heimdal kerberos support |  |
|  | [ ] OPTIMIZED_CFLAGS  Builds with compiler optimizations (-O3) |  |
|  | [X] XML            Build with XML data type (server) |  |
|  | [X] TZDATA         Use internal timezone database (server) |  |
|  | [ ] DEBUG          Builds with debugging symbols |  |
|  | [X] INTDATE        Builds with 64-bit date/time type (server) |  |
|  |                                          |  |
```

```
$ make
$ make install
$ exit
```

当编译 perl5.10 或 postgres 时，会自动跳进其他套件选项画面，
如 libiconv, m4, libxslt
我个人喜好的选择如下：

```
┌──────────────────────────────────────────────┐
```

```
|                Options for libiconv 1.13.1                  |
|  ┌──────────────────────────────────────────────────────┐   |
|  |  [X] EXTRA_ENCODINGS  Include extra character sets     |  |
|  |  [X] EXTRA_PATCHES    Apply patches to fix CP932 add EUCJP-MS  |  |


┌──────────────────────────────────────────────────────┐
|                Options for m4 1.4.13,1                |
|  ┌──────────────────────────────────────────────────┐   |
|  |    [X] LIBSIGSEGV  Use libsigsegv for better diagnostics    |  |

┌──────────────────────────────────────────────────────┐
|                Options for libxslt 1.1.26             |
|  ┌──────────────────────────────────────────────────┐   |
|  |       [X] MEM_DEBUG  Enable memory debugging          |  |
|  |       [X] CRYPTO     Enable crypto support for exslt  |  |
```

安装 postgresql-contrib

```
$ su -
$ cd /usr/ports/databases/postgresql-contrib
$ make
$ make install
$ exit
```

安装 p5-postgresql-plperl

```
$ su -
$ cd /usr/ports/databases/p5-postgresql-plperl
$ make
$ make install
$ exit
```

4. 在 pgsql.home.net 修改 postgreSQL Database 预设为中文 locate

本人使用繁体中文，选用 zh_TW.UTF-8 的 locate。

请参考 postgresql 安装说明，它有提及如何改变 locate 的方法

```
$ pkg_info -D postgresql-server-8.4.1
```

我使用增加一个 login class 的方法，去改变 locate 值，步骤为
i) 修改 /etc/login.conf

ii) 修改 /etc/rc.conf

i)

```
$ su -
$ vi /etc/login.conf
$ exit
```

==========

在档案中某空白地方增加以下描述

```
#------------------
# postgresql class
#------------------
pgsql:\
    :lang=zh_TW.UTF-8:
    :setenv=LC_COLLATE=C:\
    :tc=default:
```

修改完 /etc/login.conf 必顺执行一次 cap_mkdb 使之生效

```
$ cap_mkdb /etc/login.conf
```

ii)

```
$ su -
$ vi /etc/rc.conf
$ exit
```

==========

在档案中某空白地方增加以下描述

```
#----------------------
# postgresql settings
#----------------------
postgresql_enable="YES"
postgresql_data="/usr/local/pgsql/data"
postgresql_flags="-w -s -m fast"
postgresql_initdb_flags="--encoding=utf-8 --lc-collate=C"
postgresql_class="pgsql"
```

注意 /etc/rc.conf 中的 postgresql_class 值是对应 /etc/login.conf 名称

4. 在 pgsql.home.net 初始化 postgres 系统资料库

```
$ su
$ /usr/local/etc/rc.d/postgresql initdb
$ exit
```

5. 初始化系统资料库后，便可以第一次启动 postgresql

```
$ su
$ /usr/local/etc/rc.d/postgresql start
$ exit
```

6. 使用 psql 修改 pgsql 的密码

在 FreeBSD port 的 postgresql，它的 superuser 叫 pgsql，并不是 postgres
初始化的资料库，pgsql 点认值是没有密码，在 localhost 没有密码登入，任何人皆可以
用 pgsql 登入 postgresql 为所欲为。因此，第一件事为 pgsql 设密码，使用 psql
便可

```
$ psql -U pgsql postgres
```

==========
在 psql 画面输入命令修改密码然后离开 psql

```
postgres=# \password pgsql
postgres=#\q
```

修改密码工作还有下半部份，便是修改 /usr/local/pgsql/data/pg_hba.conf

```
$ su -
$ cd /usr/local/pgsql/data
$ vi pg_hba.conf
$ exit
```

==========
把档案中的 trust 改为 md5，强迫 postgres 必须做密码登入

```
# TYPE  DATABASE    USER    CIDR-ADDRESS        METHOD

# "local" is for Unix domain socket connections only
local   all         pgsql                   md5

# IPv4 local connections:
```

```
host    all         pgsql    127.0.0.1/32        md5
host    all         pgsql    192.168.1.3/32      md5
host    all         all     192.168.1.0/24      reject


# IPv6 local connections:
#host   all        all        ::1/128            trust
```

仔细的 pg_hba.conf 权限语法请自行参考资料

修改了 pgsql 的密码 和 pg_hba.conf 设定后，必须重新启动 postgresql

```
$ su -
$ /usr/local/etc/rc.d/postgresql restart
$ exit
```

成功了，现在登入 postgresql 必须要输入密码

6. FreeBSD postgresql periodic script

FreeBSD port 会安装了一个 periodic 在 /usr/local/etc/periodic/daily/502.pgsql
这个 periodic script 提供两个服务：

i) 每天定时做 vacuumdb 一次
ii) 每天定时做 pg_dump 一次

想启动这功能 要修改 /etc/periodic.conf
或 /etc/periodic.conf.local 二者选一，内容一样

```
$su -
$vi /etc/periodic.conf
$ exit
```

==========

档案中增加以下内容

```
# 502.pgsql
daily_pgsql_backup_enable="YES" # do backup
daily_pgsql_vacuum_enable="YES" # do vacuum
```

如果你只想做 vacuum 不做 pg_dump，内容如下

==========

档案中增加以下内容

# 502.pgsql
daily_pgsql_backup_enable="NO" # do backup
daily_pgsql_vacuum_enable="YES" # do vacuum

由于 postgresql 已改变为加密登入，使用 periodic script 必需提供 password 给 periodic，否则会执行 vacuumdb pg_dump，失败。方法是建立一个密码 .pgpass 档案把 password 传给 periodic

.pgpass 档案必需放置在 postgresql daemon 的 $HOME 才可以配合 periodic 顺利执行

检察 postgresql 的 $HOME 在那里

$su -
$more /etc/passwd | grep PostgreSQL
$ exit

==========
我的查询结果如下
---
pgsql:*:70:70::0:0:PostgreSQL Daemon:/usr/local/pgsql:/bin/sh

表示 postgresql 的 $HOME 在 /usr/local/pgsql 这里
因此把 .pgpass 放置在 /usr/local/pgsql 成为

/usr/local/pgsql/.pgpass

步骤如下:

$ su -
$ cd /usr/local/pgsql
$ touch .pgpass
$ chmod 600 .pgpass
$ chown pgsql:pgsql .pgpass
$ exit

修改 .pgpass 内容

$ su -
$ vi /usr/local/pgsql/.pgpass
$ exit

==========
.pgpass 格式有规范的，官方说明请看
http://www.postgresql.org/docs/8.4/interactive/libpq-pgpass.html
---
localhost:5432:*:pgsql:你的 pgsql 密码

最后一步，是修改 /usr/local/etc/periodic/daily/502.pgsql 使它懂得配对
正确的用户名和密码

$ su -
$ vi /usr/local/etc/periodic/daily/502.pgsql
$ exit

==========
把其中的两行参数修改

< 原本是 >
daily_pgsql_vacuum_args="-z"
daily_pgsql_pgdump_args="-b -F c"

< 修改为 >
daily_pgsql_vacuum_args="-h localhost -U pgsql -z"
daily_pgsql_pgdump_args="-h localhost -U pgsql -b -F c"

基本安装完成。

基于 Linux 系统的性能监测技术比拼和实现攻略

ChinaUnix 网友：jerrywjl

PART1：性能监测的基本概念及分类：

详细见上一期《开源时代》

PART2：各种性能监测手段在企业中部署和实现方法：

a．snmp 协议的配置以及在 Linux 下和 Windows 上的测试方法：

刚才我们已经用了不少的篇幅介绍了 SNMP 简单网络管理协议的基本原理。现在我们即将以 Red Hat 最新的企业版本 Red Hat Enterprise Linux 5 Update 2（简称 RHEL5u2）为基础来演示如何配置 SNMP 服务。

在 RHEL5u2 中提供了一个叫做 net-snmp 的 rpm 包，net-snmp 是在 IPv4 和 IPv6 上执行 SNMP 的 v1，v2 和 v3 版本协议的一组程序。需要特意说明一下的是，由于在大多数环境下针对企业应用都

会使用稳定版本的 Red Hat Enterprise Linux 操作系统，所以后面所有操作所使用的 Linux 平台也都是 RHEL，但是那些对技术体验感兴趣的用户也可以使用 Fedora 8 或者 9 来实现上述所有的操作。

在该例子中，假设服务器 192.168.1.10 是被监测的系统，我们将在其上分别配置和启用基于 v1 和 v3 版本的 snmp 服务，而另外一台主机 192.168.1.100 充当管理工作站，并且用 snmp 命令来获得被监测系统的详细信息。

在服务器 192.168.1.10 上，首先配置 v1 版本的 SNMP 协议：

挂载 DVD 安装光盘，并从光盘中安装 snmp 相关的软件包：lm_sensor，net-snmp，snmp-utils。关于 net-snmp 包的作用刚才已介绍，而至于 net-snmp-utils 主要提供了使用 snmp 协议管理网络的一系列工具。

装完所需要的软件包之后，我们可以直接修改 snmp 的主配置文件/etc/snmp/snmpd.conf 并重启服务来直接启用 SNMPv1。采用 SNMPv1 版本的重要标志之一就是使网络管理设备访问代理时需要使用基于 Community 的团体的验证方式。这里的 Community 使用默认的 public，当然也可以根据自己的需求去修改为任意一个字符串。完成之后保存该档并运行命令重启服务：

service snmpd start

chkconfig snmpd on

为了监测是否能够正确获得整个系统中每个 MIB 的 OID 值，可以运行 snmpwalk 命令以获得响应的结果，snmpwalk 命令可通过 snmp 的 GETNEXT 动作获得 MIB 树上的管理信息。

snmpwalk –v1 –cpublic 192.168.1.10

至此为止，被监测对象上的 snmp 就算配置完成。为了说明结果，我找了一个运行于 Windows 的操作系统上的利用 snmp 协议的监测软件来看看效果。在 Windows 平台上能够实现该功能的软件有很多，例如 Whatsup，Solawins 等等。这里以 Whatsup 为例，我的监测主机上操作系统选用的是 Windows Server 2003 Enterprise Edition。IP 地址是 192.168.1.100。按照图示的步骤安装 Whatsup 软件，其实秉承 Windows 软件的安装风格——一路回车即可搞定。由于我安装的是一个 30 天的免费试用版本，所以需要在启动产品的时候选择“Activate Later”，并且在“Device Discovery Method”中选择“IP Range Scan”。之后起始地址都填入被监测设备的地址 192.168.1.10，按照在/etc/snmp/snmpd.conf 档中的内容输入团体名称“public”按照下图确定扫描内容并开始扫描，扫描时间需要根据设备的数量决定。在“Action Policy Selection”中选择“Do Not Apply an Action Policy”并结束扫描。最后通过“Report View”标签选择“Device Reports”并最终获得所有设备的 Health 状况。

在众多的性能监测软件中 Whatsup 的功能相对比较强大，而且设置方便，接口友好。在很多企业的服务监测中是一个不错的选择，而且 Whatsup 的其它视图模式和功能也比较多。至于其它的例如 Solawins 等类似的软件，在配置方面的步骤基本大同小异，所以这里就不花时间详述了。

然后配置 v3 版本的 SNMP 协议：

与 v1 版本的 SNMP 协议不一样，v3 版本最重要的特征是更强的安全性，因为团体信息在网络上是以明文形式传送。因此 v3 版本不再使用团体信息来实现认证，而是采用对称或者非对称加密方式加密用户名和密码实现认证。所以安全方面自然要比 v1 版本的高很多，不过在配置方面也显然会比 v1 版本的更加麻烦。所幸的是 net-snmp-utils 工具包为我们准备了另外一个强有力的 SNMP 配置工具——net-snmp-config，因此一般用户仍然可以通过他非常方便地实现 v3 版本的 SNMP 配置。

我们先切换到光盘，由于 net-snmp-config 工具由 net-snmp-devel 包提供，所以在安装一系列依赖包包括 beecrypt，elfutils-devel，elfutils-devel-static 后，最后还是要安装 net-snmp-devel 包。之后将 snmpd 服务停止并备份其主配置文件，然后运行命令：

net-snmp-config --create-snmpv3-user -A 12345678 -X 12345678 -a MD5 -x DES admin

关于这条命令的说明：

--create-snmpv3-user [-A authpass] [-X privpass] [-a MD5 | SHA] [-x DES | AES] [username]

命令执行之后将自动建立新的配置文件 snmpd.conf，而内容也十分简单。只有用户名和权限，而关于认证方式的信息则会存储在/var/net-snmp/snmpd.conf 文件中。

最后重启 snmpd 服务，并再次用 snmpwalk 指明通过 v3 的认证方式获取 MIB 上的 OID 信息。

命令是：

snmpwalk -v3 -u admin -l auth -a MD5 -x DES -A 12345678 -X 12345678 192.168.1.10

如果要验证配置的信息是否 OK，还是可以通过 Windows 下的 Whatsup 来监测信息，步骤基本上和上例一样，只不过更改一下 SNMP 版本并填入相应的认证信息即可。

b．snmp + mrtg 实现对网络负载的监测：

上述的操作方法是利用了一些闭源的商业软件在 Windows 下进行性能监测，对于一些经费充足的企业，这种方案不乏考虑的价值。

不过在了解了 snmp 协议的基本工作原理和配置方法之后，我们来看一下利用 snmp 在 Linux 操作系统上进行监测的解决方案。提到在 Linux 上的开源方案，不得不提及一个老牌的网络流量监测工具 mrtg。

Mrtg（Multi Router Traffic Grapher,MRTG）是一个完全免费的监测网络链路流量负载的工具软件，它通过 snmp 协议从设备得到流量信息，并将流量负载以包含 PNG 格式图形的 HTML 文文件以 Web 页面显示给用户。Mrtg 能够以非常直观的形式显示流量负载，而且在工作过程中所占用的系统资源很低。

下面我们将演示如何通过 mrtg 来获得以 snmp 协议所监测到的网络流量方面的信息，为了更好地说明在企业环境中的应用，我们会通过一台运行 MRTG 的网管工作站同时获取两台被监测服务器的网络流量信息来仿真企业对多台服务器的网络流量监测方式。

基本结构如下：

网管工作站：RHEL5u2 192.168.1.10

被监测主机 1：RHEL5u2 192.168.1.100

被监测主机 2：RHEL5u2 192.168.1.200

首先在被监测主机 192.168.1.100 和 192.168.1.200 上分别配置并开启 snmpd 服务（过程同上例）。为了简化配置我只使用上面的 v1 版本的 SNMP 配置方法。同时需要在开启服务之后关闭防火墙以及保证主机之间的连通性。

之后在网管工作站上安装并且配置 mrtg。由于在 RHEL5u2 中 mrtg 是系统自带的软件包，所以可直接使用 rpm 安装。

rpm -ihv mrtg-2.14.5-2.i386.rpm

安装完成之后需要运行命令 cfgmaker 针对两台被监测主机各自生成 mrtg 的配置文件，在该例子中配置文件分别是 test1.cfg 和 test2.cfg，存放在/etc/mrtg 目录中。

#cfgmaker --global "WorkDir: /var/www/html/mrtg" \

> --global "Options[_]: growright,bits" \

> --ifref=ip \

> --output /etc/mrtg/test1.cfg \

> public@192.168.1.100

#cfgmaker --global "WorkDir: /var/www/html/mrtg" \

> --global "Options[_]: growright,bits" \

> --ifref=ip \

> --output /etc/mrtg/test2.cfg \

> public@192.168.1.200

上述的命令定义了生成配置文件 test1.cfg 和 test2.cfg 的全局参数，包括配置文件的主目录，页面存放的主目录，snmp 团体信息和建立绘图时指定绘图方式的一些必须参数，如绘制向右方增长的统计图和统计图的计量单位等。

之后执行下面的命令将两个配置文件的内容合并到主配置文件/etc/mrtg/mrtg.cfg 里面。

cat test1.cfg >> mrtg.cfg

cat test2.cfg >> mrtg.cfg

并根据配置文件的需求在/var/www/html 目录下建立 mrtg 页面的主目录：

mkdir /var/www/html/mrtg

以及针对 mrtg.cfg 配置文件运行命令来启动 mrtg，注意，在默认的 UTF-8 语言字符集下这个启动命令无法执行成功，因此需要设置语言环境变量为 env=C：

[root@localhost mrtg]# env LANG=C /usr/bin/mrtg /etc/mrtg/mrtg.cfg

该命令需要执行三次以建立正确的 mrtg 数据库文件。

最后要做的工作是按照配置文件内容在 mrtg 页面的主目录下生成正确的 index 檔，命令如下：

indexmaker --output /var/www/html/mrtg/index.html \

> --title=MRTG \

> /etc/mrtg/mrtg.cfg

同时也要按照 mrtg.cfg 的配置修改和启动 apache 并最终为 mrtg 能够定期进行数据采集建立一个每五分钟执行一次的任务计划：

```
# cat /etc/httpd/conf/httpd.conf | grep DocumentRoot
DocumentRoot "/var/www/html/mrtg"

# service httpd start
# chkconfig httpd on

# crontab -l
*/5 * * * * /usr/bin/env LANG=C /usr/bin/mrtg /etc/mrtg/mrtg.cfg /dev/null 2>&1
```

在启动 apache 之后，即可以在网管工作站或者是任何可以访问到 192.168.1.10 的主机上以 http://192.168.1.10 的方式打开 MRTG 的页面。可以很直观地看到两台主机的流量页面以及详细信息。

c．SYSSTAT + mrtg 实现对服务器各种性能参数的监测：

一些读者可能会有这样的疑虑：既然我们已经能够通过 snmp + mrtg 得到网络流量的监测情况，那么如果要实时监测服务器的其它系统参数应该怎么办？这个问题的答案比较遗憾：由于 MRTG 是一个专门针对网络流量进行监测和绘图的工具，所以默认情况下不提供对系统其它方面信息监测的功能。因此尽管 SNMP 协议本身可以获得和显示被监测主机上的大量信息但是鉴于 MRTG 方面的限制而无法显示和显示出来。

但是好在天无绝人之路，MRTG 本身实际上也是一个强大的数据采集和绘图引擎。于是我们可以利用一些 SNMP 以外的系统监测工具来实时获取服务器性能信息，包括 CPU，内存，磁盘空间使用率以及 I/O 性能方面的内容，然后交给 MRTG 来获得我们所需要的其它类型的统计图信息。

在清楚了 MRTG 原理之后，这样做实际上会拥有更大的灵活性。但是由于涉及的内容需要一定脚本编程的知识，所以对一些高级用户是比较适用的。

下面我们就举例说明如何将 MRTG 和 SYSSTAT 所提供的一系列如 sar，iostat 以及 free 等性能监测命令等进行结合来获得系统其它方面的统计信息的配置方法。

由于这次网管工作站已经不可能再通过 SNMP 获取信息，所以我们将环境更改一下，在被监测主机直接安装和配置 MRTG，并且结合 sysstat 一类的系统工具来绘制本机的信息图：

此时假设被监测主机的操作系统是：RHEL5u2，IP 地址是 192.168.1.100

Sysstat 是在 RHEL5u2 中自带的一个综合性能监测工具包。其中包括了 sar（主要用于 cpu，内存方面的信息统计）和 iostat（存储设备 I/O 统计工具）。由于 sysstat 是系统本身自带的包，所以在两台被监测的主机上分别挂载光盘安装 sysstat 即可。当然除了 sysstat 还有一些其它的系统性能显示方面的工具可以使用，我们会分别举例。

```
# rpm -ihv sysstat-7.0.2-1.el5.i386.rpm
```

同时还要分别在被监测的主机上安装 mrtg。

```
# rpm -ihv mrtg-2.14.5-2.i386.rpm
```

在正式开始之前，需要花一点时间来介绍一下 sar 和 iostat 的基本功能：sar 是一个强大的系统监测工具，默认只显示 CPU 的使用情况，而通过加上不同的参数 sar 可以显示大量的内存以及 I/O 使用方面的状况和信息；而 iostat 主要显示 I/O 使用方面的信息。如果执行 sar 不加任何参数通常会显示下面这些信息：

其中 sar 的命令显示内容包括了：

%user　　用户空间的进程占用 CPU 时间的百分比

%system　 系统空间的进程占用 CPU 时间的百分比

%nice　　已调整优先级的进程占用 CPU 时间的百分比

%iowait　没有进程在该 CPU 上执行时，处理器等待 I/O 完成的时间

%steal　　虚拟操作系统占用 CPU 时间的百分比

%idle　　没有进程在该 CPU 上执行的时间（也就是 CPU 未使用的时间）

其中 iostat 命令显示内容包括了：

tps　　每秒钟完成的 I/O 请求数量

Blk_read/s　　每秒从设备上读取的 block 数量

Blk_write/s　　每秒写入到设备商的 block 数量

另外 sar 可以通过增加时间参数来指定执行的频率以及输出的内容量，例如命令是 sar －u 5，则表示每 5 分钟显示一次 CPU 的使用情况。这样就利于我们将该命令写入到一个脚本/var/www/mrtg/mrtg.cpu 中。在执行之前需要首先赋予其执行权限，然后为了监测该脚本是否有语法错误，可以执行该脚本测试。如果显示信息正确，则可以将其嵌入到自建的监测 CPU 的 mrtg 配置文件 mrtg.cfg.cpu 中。当然这个时候本机 Apache 的访问主目录也要更改为/var/www/html/mrtg 并重启 apache 服务。之后运行命令三次以启动 mrtg：

env LANG=C /usr/bin/mrtg /var/www/mrtg/mrtg.cfg.mem

最后别忘了建立存放页面的目录，进入该目录中，将 localhost.html 更名为 index.html 以及类似上面的例子增加一个每五分钟运行一次的计划任务。此时在任何一台工作站上通过浏览器访问页面 http://192.168.1.100/cpu 就可以看到 192.168.1.100 的 CPU 使用情况。

按照上面的方法，对内存使用情况的监测也大同小异：

可以使用 sar －r 选项来显示内存使用情况。同样将该命令写入到一个脚本/var/www/mrtg/mrtg.mem 中去，并给予执行权限。在测试执行正常之后即可将其嵌入到自建的监测内存的 mrtg 配置文件 mrtg.mem.cfg 中，执行命令# env LANG=C /usr/bin/mrtg /var/www/mrtg/mrtg.cfg.mem 三次，并建立存放页面的目录，将 localhost.html 更改为 index.html，并建立类似上面的计划任务。完成之后从任何一台工作站通过浏览器都可以访问页面 http://192.168.1.100/mem 来查看该主机的内存使用情况。

接着按照上面的例子，来建立磁盘读写情况的监测：

由于原理一致，方法接近。我就不再赘述，只是会将基本的过程和需要的脚本内容给出来：

cd /var/www/mrtg/

建立监测脚本及其内容：

vi mrtg.disk

cat mrtg.disk

```
#!/bin/bash
hd=sda
disk=/dev/$hd
UPtime=`/usr/bin/uptime |awk '{print $3""$4""$5}'`
KBread_sec=`iostat -x $disk|grep $hd |awk '{print $8}'`
KBwrite_sec=`iostat -x $disk|grep $hd |awk '{print $9}'`
echo $KBread_sec
echo $KBwrite_sec
echo $UPtime
hostname
```

赋予权限：

chmod 755 mrtg.disk

测试执行情况：

```
# ./mrtg.disk
37.51
0.16
7:25,3users,
localhost.localdomain
```

建立配置文件及显示其内容：

vi mrtg.cfg.disk

cat mrtg.cfg.disk

```
WorkDir: /var/www/html/mrtg/disk
Target[disk]: `/var/www/mrtg/mrtg.disk`
Title[disk]: Disk HDA I/O Utilization Report
#Unscaled[disk]: dwym
MaxBytes[disk]: 10240000
```

PageTop[disk]: <H1>Disk I/O Utilization Report</H1>

kmg[disk]: KB,MB,GB

LegendI[disk]: Disk I/O KBread/sec

LegendO[disk]: Disk I/O KBwrite/sec

Legend1[disk]: Disk I/O KBread/sec

Legend2[disk]: Disk I/O KBwrite/sec

YLegend[disk]: Megabytes

ShortLegend[disk]: &

Options[disk]: growright,gauge,nopercent

建立主页面主目录:

[root@localhost mrtg]# mkdir /var/www/html/mrtg/disk

运行 mrtg:

[root@localhost mrtg]# env LANG=C /usr/bin/mrtg /var/www/mrtg/mrtg.cfg.disk

[root@localhost mrtg]# env LANG=C /usr/bin/mrtg /var/www/mrtg/mrtg.cfg.disk

[root@localhost mrtg]# env LANG=C /usr/bin/mrtg /var/www/mrtg/mrtg.cfg.disk

重命名 index 檔:

[root@localhost mrtg]# mv /var/www/html/mrtg/disk/disk.html /var/www/html/mrtg/disk/index.html

建立任务计划:

[root@localhost mrtg]# crontab -l -u root

*/5 * * * * /usr/bin/env LANG=C /usr/bin/mrtg /var/www/mrtg/mrtg.cfg.cpu /dev/null 2>&1

crontab: installing new crontab

重启 httpd 服务:

[root@localhost mrtg]# service httpd restart

[root@localhost mrtg]# chkconfig httpd on

完成之后从任何一台工作站通过浏览器都可以访问页面 http://192.168.1.100/disk 来查看该主机的磁盘读写效率情况。

到此为止我们基本上对 mrtg 在各方面的功能已经有了一个比较全面的认识。而且针对其配置和管理方法也有了一个初步的了解。相信在企业中会有更多的朋友能够藉助手工定制的方式来灵活发掘和获取 mrtg 在其它方面更多的功能和特性。

d．更加强大和灵活的性能监测方案：SNMP + Cacti + RRDtool

通过对 MRTG 这个软件的使用很多用户不难发现。像 MRTG 这样的软件尽管有系统资源占用少和低

成本等方面的优点，其实也存在着一些功能方面的限制：

例如 MRTG 本身只是通过 SNMP 协议来监测网络流量的一个工具软件，所以在设计上并没有考虑到提供足够的和 SNMP 协议结合的功能以实现对服务器其它方面的性能参数进行监测。简单的说，SNMP 获得的信息尽管足够全面，但是 MRTG 默认的配置方式只能对其网络信息作监控，这样对于 SNMP 协议来说就有点大材小用之嫌。

所以如果要实现对 SNMP 所获得的更多信息进行统计就必须要像上面那样手动定制脚本并将获取的数据指定到 MRTG 配置文件上。这实际上只是利用 MRTG 来做一个信息采集与绘图的软件。这种操作给我个人的感觉似乎有些不伦不类，况且手动定制脚本尽管的确可以拥有一定的灵活性，但这对于一些初级用户来说还是存在一些技术上的困难。另外如果这样的结构扩展到拥有多个服务器的网络中，就需要在每一个服务器上都要部署同样的架构来实现多台服务器同步监测，显然工作量就显得比较大了。

另外 mrtg 的数据库是一种文本式的数据库，一般数据不能重复使用，只能画出两个 DS（一条线，一个块）并且缺乏管理功能。

能否有一款方案能够完全利用 SNMP 协议中众多的 MIB 和 OID 信息绘制出内容综合全面而且接口美观的系统性能分析图表呢？答案就是现在即将出场的 RRDtool 和 Cacti。

RRDtool 其实和 mrtg 是同一家族，主要目的都是产生 time-series 的图文件(如流量，负载，温度.....)。不过因为 mrtg 当初的考虑是画两种资料在图上(或四个值)，原作者觉得这样的功能十分不足，所以后来另外又开发了 Rrdtool。

RRDtool 本身可和 mrtg 结合，但其结合基本上仅在于将 mrtg 的文字文件的 log 转成 rrd 储存格式，使用 rrd 存储格式，数据能重复使用，例如可以将一个 rrd 文件中的数据与另一个 rrd 文件中的资料相加；可以定义任意时间段画图，即你可以画出一张半年以来的数据的图，也可以画出一张半小时以来的图；能画任意个 DS；CDEF 也可以任意定制。所以 RRDtool 变成了几乎最终也是最好的选择。但由于 RRDTool 的指令非常复杂，对于使用者来说显得非常的麻烦，而且 RRDtool 作为一个强大的绘图引擎缺少 mrtg 那种数据采集功能。

幸运的是有一套软件 Cacti 的发展就是基于让 RRDtool 使用者更方便使用该软件，除了基本的 SNMP 流量跟系统信息监测外，Cacti 也可外挂 Scripts 及加上 Templates 来作出各式各样的监测图。

Cacti 其实是一套 php 程序，它运用 snmpget 采集数据，使用 RRDtool 绘图。使用 Cacti 能统计网络设备的流量、CPU、系统负载等参数，也可以自定义监测的指标。它的界面非常漂亮，能让你根本无需明白 RRDtool 的参数能轻易的绘出漂亮的图形。更难能可贵的是，它提供了强大的数据管理和用户管理功能，一张图是属于一个 host 的，每一个 host 又可以挂载到一个树状的结构上。

RRDtool 与 Cacti 配合的工作流程如下：

Cacti 会通过 SNMP 定时采集并存储被监测设备的各种数据信息，而这些数据信息会被以 rra 文件形式存储在 Mysql 数据库中。如果当用户需要查询这些数据信息的时候会通过 Cacti 将请求提交给 Mysql 数据库来查找设备对应的 rra 文件名称，并同时通过 RRDtool 绘制流量图以及返回给用户。

用户的管理上，作为一个开源软件，它可以做到为指定一个用户能查看的“树”、host、甚至每一张图，还可以与 LDAP 结合进行用户的验证，可以说，Cacti 将 RRDtool 的所有“缺点”都补足了！

有关于 Cacti 和 RRDtool 方面的更多信息可以访问其官方网站：

http://www.cacti.net/

http://oss.oetiker.ch/rrdtool/

下面我们将通过一个实际的例子讲解如何在企业中部署 SNMP + Cacti + RRDtool。

操作的环境：

监测主机 IP 地址是 192.168.1.200，RHEL5u2 系统，这里也称为 Server 或者网管主机（工作站）

被监测主机 IP 地址是 192.168.1.220，RHEL5u2 系统，这里也称为 Client

由于这些都是系统光盘中自带的 Red Hat 官方提供的包。所以挂在光盘后执行 rpm –ihv 就可以逐一装上。

在安装完成之后，配置并开启本机的 SNMP。简单起见我仍然只开启 V1 版本的 SNMP，由于在上面说明 SNMP 配置原理的时候已经有了具体步骤，所以这里就不再赘述。

在 SNMP 服务启动之后可以下载和编译 rrdtool 以及 cacti。

从以下站点下载这两个软件：

cacti：

http://www.cacti.net/downloads/cacti-0.8.7b.tar.gz

rrdtool：

http://oss.oetiker.ch/rrdtool/pub/rrdtool-1.2.28.tar.gz

目前这两个软件的版本都是最新版本。

首先将这两个软件拷贝到/usr/local 目录下，然后分别按照下面的步骤对其进行解压，编译以及安装。

```
# cd /usr/local/
# ls rrdtool-1.2.28.tar.gz
rrdtool-1.2.28.tar.gz
# ls cacti-0.8.7b.tar.gz
cacti-0.8.7b.tar.gz
# cd rrdtool-1.2.28
# ./configure --prefix=/usr/local/rrdtool
# make
# make install
```

为了保证编译能够顺利进行，需要预先在系统中安装 gcc 编译器等开发工具包。

至于 cacti，将其解压后把整个目录拷贝到/var/www/html 目录下并重命名为 cacti。

下面我们需要为 cacti 创建所需的数据库，这也是当时安装 mysql-server 的原因。

先启动 mysql 服务，并且指定 mysql 管理员的账号以及密码，然后以新的用户名和密码重新登录 mysql 数据库。

下面的操作是在 mysql 的交互接口下进行。首先需要通过 T-Sql 语句建立 cacti 数据库，然后为方便管理，需要给 root 以及 cactiadmin 用户授予所有管理权限，并且定义 cactiadmin 用户的密码。完成之后退出交互接口，并且将 cacti 预先定制的 cacti.mysql 文件导入到 mysql 中去。

这时再通过交互接口进入 mysql 之后就可以看到刚才所导入的 cacti 预先定制好的表了。

接着按照刚才导入到数据库中的内容用 vi 编辑器修改 cacti 管理接口的配置文件：

vi /var/www/html/cacti/include/config.php

修改的内容如下：

$database_type = "mysql";

$database_default = "cacti";

$database_hostname = "localhost";

$database_username = "cactiadmin";

$database_password = "123456";

$database_port = "3306";

由于我们需要定制 cactiadmin 通过 SNMP 定时到被监测设备中获取数据信息，所以要建立 cactiadmin 用户并以该用户的身份建立一个任务计划。

useradd cactiadmin

passwd cactiadmin

crontab -e -u cactiadmin

crontab -l -u cactiadmin

*/5 * * * * /usr/bin/php /var/www/html/cacti/poller.php > /dev/null 2>&1

同时指定下面两个目录的属主是 cactiadmin

chown -R cactiadmin /var/www/html/cacti/rra /var/www/html/cacti/log

之后我们需要安装一个叫做 cactid 的东西，这个所谓的 cactid 是 cacti 的数据收集器的守护进程。

安装和配置的方法也很简单。

最新的 cactid 版本是 0.8.6k，可以从该地址获得：

http://mirrors.rootservices.net/cacti/cactid/cacti-cactid-0.8.6k.tar.gz

下载之后将其放到/usr/local 目录下：

然后按照下面步骤进行编译安装：

tar – zxf cacti-cactid-0.8.6k.tar.gz

cd cacti-cactid-0.8.6k

./configure --prefix=/usr/local/cactid

make

make install

为了保证编译成功，还需要下面的几个软件：

mysql-devel

libpng

libpng-devel

zlib

zlib-devel

freetype

freetype-devel

在该软件编译完成之后，在/usr/loca/目录下会自动建立 cactid 目录，进入该目录修改 etc/cactid.conf 档，修改之后的结果如下：

DB_Host localhost

DB_Database cacti

DB_User cactiadmin

DB_Pass 123456

DB_Port 3306

并且将该文件复制到/etc 目录下。

cp /usr/local/cactid/etc/cactid.conf /etc/

最后配置 HTTP 服务，只需要将网站服务的主目录指向/var/www/html/cacti 然后重启 httpd 服务即可：

DocumentRoot "/var/www/html/cacti"

完成之后可以在监测主机上开启一个浏览器并且直接访问 http://192.168.1.200，由于已经将 cacti 设置为网站的主目录，所以上述的配置一切正常的话可以获得一个 cacti 的初始安装页面。点击“Next”之后开启下一个页面准备安装 cacti。在出现的页面中如果确认所有信息无误则选择“New Install”并进入下一步。在新出现的 Installation guide 页面中 cacti 会自动探测原先安装到一些基本文件的位置。在这里需要确保安装程序能够找到所需要的所有文件和信息。针对第一个地方出现的“RRDtool binary path not found”的情况，需要手动更改一下路径为：/usr/local/rrdtool/bin/rrdtool，并点击“Finish”。

在接着出现的新页面中输入用户名和密码。此处默认的用户名和密码都是 admin。进入之后 cacti 会强制用户更改密码，为了简化起见，我这里仍然更改为 123456 并保存密码。在当前页面点击

console 卷标之后点击 setting 按钮并选取 path 卷标，在该页面上需要指定“Cactid Poller File Path”，这里就是 cactid 程序的绝对路径/usr/local/cactid/bin/cactid，同时确认其它的 RRDtool required path 的设定是正确的，保存该配置。到此为止，整个 cacti 主页面的全局设置就基本完成，最后开启添加要监测的目标服务器。点击“Console”然后点击“Create”标签下面的“New Graphs”按钮以及在出现的页面上点击“Create New Host”进入建立被监测服务器的页面，按照下面的页面来添加被监测系统的信息，其中包括：描述（可随便填写），主机名称（IP 地址），监测的主机类型（Local Linux Machine），SNMP 版本（V1）和与之对应的埠号（161）与团体信息。最后点击“Create”。

但是由于我使用的是 RHEL5u2 系统上自带的 firefox 3 浏览器，在这个时候会出现一些问题。如图，提示“由于密码不符，该设备无法建立”，但事前我们并没有设置过任何密码。这其实是在该版本的 cacti 上的一个 bug 而已，在 RHEL 系统上尤其针对比较新的 firefox3 浏览器，当提交到该页面的时候由于一些缓存方面的影响而导致无法最终完成该步骤。不过解决的方法也很简单。就是使用 IE 系列浏览器或者版本比较低的 firefox 浏览器来访问，那么在做到该步骤的时候就可以正常完成配置。

在该步骤完成之后可以看到默认的配置将会监测目标主机的内存、负载、登录用户以及 CPU 方面的信息。而用户完全可以在“Add Graph Template”以及“Add Data Query”处添加更多的监测类型。并选择右边的“Add”，最后将配置保存并切换到“Graph”标签下复选需要监测的参数类型，一般着重选择的内容包括 CPU，内存，Swap 分区，Buffer 缓冲区，网络流量，磁盘使用量以及登录用户等。完成之后在“Graph”的“Preview”卷标中即可获得动态的统计图信息。由于我选的内容太多，所以在当前页面无法全部显示出所有统计图来，只能一个个项目点击进去。

和 Mrtg 类似，每一个项目的监测统计都有 5 分钟、30 分钟、2 小时和 1 天为计算单位。所以从各种统计图来看，即可获得一个参数当前的信息，又可获得相对长久的信息便于比较。

而作为被监测主机，其实需要配置的内容只有两个：第一，修改 SNMP 配置文件并启动服务，使之与网管主机的 SNMP 版本和认证方式一致；第二，确保网络连通性以及确保网管主机能够通过 snmpwalk 获得被监测主机上的全部 MIB 内容。由于配置方法和上面的例子一样，只要从监测主机上将其已经修改过的 snmpd.conf 配置文件拷贝到/etc/snmpd 目录下覆盖原有配置文件，然后启动 snmpd 服务并关闭防火墙就行了。当然可以分别在监测主机和被监测主机上运行命令来测试：

snmpwalk –v1 –cpublic 192.168.1.220

如果能够获得所有的 MIB 信息，则证明客户端的配置没有问题，所以这里也不再赘述。

到此为止 Cacti + RRDtool 的结构就已经配置完成，相对于 Mrtg 而言，Cacti + RRDtool 的方式有一个更大的好处就是利于用户时刻调整监测的对象。例如，用户需要调整当前的监测内容并新增一些监测项目。那么就可以随时编辑监测设备并修改需要监测的参数，相当方便。同时 Cacti + RRDtool 提供了在 Mrtg 上不具备的管理功能，以及 Cacti 的绘图功能相对更强大，图像更加美观。

尽管第一次部署网管主机比较复杂，但是后续对配置对多个设备的集中监测就相对简单多了。可以说是一劳永逸。

这些都是这种结构尽管在配置方面略显复杂，但是越来越受欢迎并在企业中广泛应用的原因。和 Mrtg 一样，Cacti 和 RRDtool 也是一个目前在企业中越来越广泛应用的开源监测方案。

e．企业级性能监测方案：Nagios

在大多数情况下 Cacti + RRDtool 已经实现对系统各种参数的监测。但一些要求更高更严格的企业可能不满足于仅仅监测系统基本参数的需求，而是需要监测除基本参数之外的各种应用程序的运行状况。很显然在这种情况下对于一些系统或者是自定义的程序 Cacti + RRDtool 的局限性就显示出来了。而此时就轮到了另外一种监测系统的登场。这就是我们现在要介绍的 Nagios。

Nagios 是一个功能非常强大的开源的系统网络监测程序，通过访问 http://www.nagios.org 可以了解其基本特性。Nagios 不但能够实现对系统 CPU，磁盘、网络等方面参数的基本系统监测，而且还能够监测包括 SMTP，POP3，HTTP，NNTP 等各种基本的服务类型。另外通过一些插件的安装和监测脚本自定义用户可以针对自己的应用程序实现监测，并针对大量的监测主机和多个对象部署层次化的监测架构。而且在监测信息统计方面，Nagios 也能够和例如 Cacti 等程序结合来提供动态统计图表。除此之外 Nagios 拥有强大的日志管理系统，可以实现详细的日志记录以及回卷。最难能可贵的是 Nagios 提供了优秀的事件报警功能，能够将一些突发的事件以电子邮件的形式通知管理员并能够针对出现的问题提供一些主动的解决建议和方案，并支持冗余监视。

相对于 Mrtg 以及 RRDtool + Cacti 而言 Nagios 最大的特点之一是其设计者将 Nagios 设计成监测的管理中心尽管其功能是监测服务和主机，但是他自身并不包括这部分功能的代码，所有的监测、监测功能都是由相关插件来完成的，包括报警功能。Nagios 自身也没有报警部分的代码和插件，而是交给用户或者其它相关开源项目组去完成。对于 Nagios 这个监测中心来说，细致的工作必然是交给其它的软件来实现。

下面我们就开始实施 Nagios 的基本安装和配置。

我的操作环境是：

监测主机：IP：192.168.1.10 操作系统：RHEL5u2

被监测主机：IP：192.168.1.220　　操作系统：RHEL5u2

Nagios 的所有软件包可以从其官方网站获得 http://www.nagios.org/download/。这里无论是基本软件包还是插件我都使用的最新版本。因为我使用的操作系统是 Red Hat 最新版本，所以自然要好马配好鞍。

首先在监测主机也就是 192.168.1.10 上安装 Nagios 的基本软件包，在安装 Nagios 之前首先需要保证系统中有下面这些软件包：Apache，gcc，gd，gd-devel，glibc，glibc-devel。可以用 rpm –qa | grep 的方式去逐一检查。

如果确认上面这些包都安装之后需要先建立 Nagios 的用户和 nagcmd 组：

useradd -m nagios

passwd nagios

并将 nagios 以及 apache 用户加入到 nagcmd 组中

groupadd nagcmd

usermod -G nagcmd nagios

usermod -G nagcmd apache

完成之后将下载的 nagios 压缩包拷贝到/usr/local 目录中，并且执行下面的步骤进行编译和安装：

tar -zxf nagios-3.0.3.tar.gz

cd nagios-3.0.3

首先初始化和建立编译的环境

./configure --with-command-group=nagcmd

如果能看到下面的基本配置信息则说明初始的环境已经成功配置完成：

*** Configuration summary for nagios 3.0.3 06-25-2008 ***:

General Options:

Nagios executable: nagios

Nagios user/group: nagios,nagios

Command user/group: nagios,nagcmd

Embedded Perl: no

Event Broker: yes

Install ${prefix}: /usr/local/nagios

Lock file: ${prefix}/var/nagios.lock

Check result directory: ${prefix}/var/spool/checkresults

Init directory: /etc/rc.d/init.d

Apache conf.d directory: /etc/httpd/conf.d

Mail program: /bin/mail

Host OS: linux-gnu

Web Interface Options:

HTML URL: http://localhost/nagios/

CGI URL: http://localhost/nagios/cgi-bin/

Traceroute (used by WAP): /bin/traceroute

Review the options above for accuracy. If they look okay,

type 'make all' to compile the main program and CGIs.

之后按照提示执行命令来进行编译：

make all

如果编译过程顺利完成，则需要执行下面的命令：

make install

make install-init

make install-config

make install-commandmode

分别用于安装二进制文件、初始化脚本、示例配置文件和设置目录权限。

ls /usr/local/nagios

安装完成之后，在/usr/local/nagios 目录下如果能够看到这些目录：bin etc sbin share var 就表示 Naigos 安装成功了。

不过在完成之后还不能启动 Nagios，因为还有一些操作需要执行。

Nagios 的样例配置文件默认安装在/usr/local/nagios/etc 目录下，这些样例文件可以配置 Nagios 使之正常运行，只需要做一个简单的修改。用你擅长的编辑器软件来编辑这个 /usr/local/nagios/etc/objects/contacts.cfg 配置文件，更改 email 部分，在 nagiosadmin 的联系人定义信息中的 EMail 信息为你的 EMail 信息以接收报警内容。

vi /usr/local/nagios/etc/objects/contacts.cfg

之后执行下面的命令来安装 Nagios 的 WEB 配置文件到 Apache 的 conf.d 目录下：

make install-webconf

在 Apache 中使用基本认证的方式创建一个 nagiosadmin 的用户用于 Nagios 的 WEB 界面登录。记下你所设置的登录口令。该用户登录口令和账号信息会存储 到/usr/local/nagios/etc/passwd.users 文件中

htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin

在 Nagios 主程序安装之后会自动将相关 apache 配置文件放到/etc/http/conf.d 目录下，文件名是 nagios.conf。文件内容如下：

cat /etc/httpd/conf.d/nagios.conf

```
ScriptAlias /nagios/cgi-bin "/usr/local/nagios/sbin"


<Directory "/usr/local/nagios/sbin">
  Options ExecCGI
  AllowOverride None
  Order allow,deny
  Allow from all
```

```
  AuthName "Nagios Access"
  AuthType Basic
  AuthUserFile /usr/local/nagios/etc/htpasswd.users
  Require valid-user
</Directory>

Alias /nagios "/usr/local/nagios/share"

<Directory "/usr/local/nagios/share">
  Options None
  AllowOverride None
  Order allow,deny
  Allow from all
  AuthName "Nagios Access"
  AuthType Basic
  AuthUserFile /usr/local/nagios/etc/htpasswd.users
  Require valid-user
</Directory>
```

这就意味着只有通过认证用户才可以通过 http 访问/usr/loca/nagios/share 以及/usr/local/nagios/sbin 目录下内容。也就是 nagiosadmin，之后可以重启 apache 来应用配置：

```
# service httpd restart
# chkconfig --level 345 httpd on
```

刚才已经提到 Nagios 主程序只是一个控制中心，而能够起到服务监测和系统监测等功能的是 Nagios 插件，没有插件的 Nagios 系统只是一个空壳。因此在安装了 Nagios 平台之后需要安装插件。

Nagios 插件同样是在其官方网站下载，目前版本是 1.4.12。我将下载的源码包放到/usr/local 目录下，按照下面的步骤进行解压，编译和安装：

```
# tar -zxf nagios-plugins-1.4.12.tar.gz
# cd nagios-plugins-1.4.12
# ./configure --with-nagios-user=nagios --with-nagios-group=nagios
# make
```

make install

然后把 Nagios 加入到服务列表中以使之在系统启动时自动启动：

chkconfig --add nagios

chkconfig nagios on

执行下面的命令来验证 Nagios 的样例配置文件：

/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg

如果最后的结果类似下面而没有报错，可以启动 Nagios 服务：

Total Warnings: 0

Total Errors: 0

Things look okay - No serious problems were detected during the pre-flight check

service nagios start

之后可以在浏览器上访问 http://192.168.1.10/nagios，如果能够正常看到页面，证明主程序和插件都安装和配置成功！点击“Service Detail”的链接来查看你本机的监视详情。此时可能需要给点时间让 Nagios 来检测你机器上所依赖的服务。

实际上在装完 Nagios 之后此时网络监控工作只是刚开始，毫无疑问用户的需求不是只监控本地系统，而是大量的远程服务器上的系统状况以及服务运行状况。

有几种不同方式来监控远程 Linux/UNIX 服务器的服务与属性。一个是应用共享式 SSH 密钥运行 check_by_ssh 插件来执行对远程主机的检测。这种方法会导致安装有 Nagios 的监控服务器产生很高的系统负荷，尤其是要监控成百个主机中的上千个服务时，这是因为要建立 SSH 连接的总开销很高。

另一种方法是使用 NRPE 外部构件监控远程主机。NRPE 外部构件可以在远程的 Linux/Unix 主机上执行插件程序。如果是要象监控本地主机一样对远程主机的磁盘利用率、CPU 负荷和内存占用率等情况下，NRPE 外部构件将非常有用。

提到“外部构件”这个概念的时候需要说明一下，Nagios 有许多"外部构件"软件包可供使用。外部构件可以扩展 Nagios 的应用并使之与其它软件集成，而且能够通过 WEB 接口来实现管理配置文件，监控远程主机（*NIX，Windows 等），对远程主机的强制监测，减化并扩展告警逻辑等功能。

NRPE 是一个可在远程 Linux/Unix 主机上执行的插件的外部构件包。如果你需要监控远程的主机上的本地资源或属性，如磁盘利用率、CPU 负荷、内存利用率等时是很有用的。像是用 check_by_ssh 插件来实现的功能一样，但是它不需要占用更多的监控主机的 CPU 负荷，所以当你需要监控大量的主机时这个构件将起到很重要的作用。通过该图可以看出，我们需要在被监测主机上部署 NRPE，他相当于一个 daemon 负责监听。而监测主机使用 check_nrpe 并通过 SSL 连接访问这个 daemon，然后调用被监测方的 check_disk，check_load 等脚本获取信息并将结果传递到监测主机。同时这些脚本也有能力监测到其它主机的相关信息。

NRPE 的使用环境有 direct check 和 indirect check 两种，direct check 指的是 NRPE 运行在被监测主机的本地，而 indirect check 意味着运行 NRPE 的服务器只是一个中间人，他会继续通过刚才所提及的脚本来监测其它更多远程主机上的服务和系统信息。层次化的监测就是通过这种方式来实

现。

为了简单说明问题，我们将要部署基于 direct check 的环境部署 NRPE。所以下面的操作在被监测主机 192.168.1.220 上进行。

首先要建立 Nagios 账号，这里我使用同样的密码：

useradd nagios

passwd nagios

之后按照和上面相同的步骤来编译和安装 nagios-plugin 软件：

tar -zxf nagios-plugins-1.4.12.tar.gz

cd nagios-plugins-1.4.12

./configure

make

make install

然后对相关的目录设置权限和所属用户组：

chown nagios.nagios /usr/local/nagios

chown – R nagios.nagios /usr/local/nagios/libexec

接着 NRPE 包放到/usr/local 目录下，按照下面的步骤解压缩，并且编译和安装：

tar -zxf nrpe-2.12.tar.gz

cd nrpe-2.12

./configure

make all

make install-plugin

make install-daemon

make install-daemon-config

同时安装 NRPE 的插件、进程以及进程范例配置文件。

接着执行命令将 nrpe 安装为依赖 xinetd 超级进程的非独立服务，那么前提必须安装 xinetd。不过一般系统都会自动安装该服务。

最后执行下面的命令将 NRPE 安装为 xinetd 超级进程所管理的进程之一。

make install-xinetd

完成之后需要编辑/etc/xinetd.d 目录下的 nrpe 文件，并且在最后添加允许实施监测的主机 IP 地址，这里是 192.168.1.10，那么配置文件全文如下：

cat /etc/xinetd.d/nrpe

service nrpe

```
{
        flags           = REUSE
        socket_type     = stream
        port            = 5666
        wait            = no
        user            = nagios
        group           = nagios
        server          = /usr/local/nagios/bin/nrpe
        server_args     = -c /usr/local/nagios/etc/nrpe.cfg --inetd
        log_on_failure  += USERID
        disable         = no
        only_from       = 192.168.1.10
}
```

然后修改/etc/services 档，并添加下面的内容：

```
nrpe            5666/tcp                        # nrpe
```

重启服务：

```
# /etc/init.d/xinetd restart
```

此时检查 nrpe 服务启动状况如下：

```
# netstat -nl | grep 5666
tcp        0      0 0.0.0.0:5666                0.0.0.0:*                   LISTEN
# lsof -i:5666
COMMAND  PID USER   FD   TYPE DEVICE SIZE NODE NAME
xinetd  9949 root    5u  IPv4  28764       TCP *:nrpe (LISTEN)
```

现在最关键的一步是确保安装的 NRPE 进程能够正常工作，所以要使用 check_nrpe 插件进行测试。在监测主机 192.168.1.10 上执行命令：

```
# /usr/local/nagios/libexec/check_nrpe -H 192.168.1.220
```

如果能够出现如下的版本号显示，则证明在被监测主机上配置的 NRPE 已经正常工作，并且监测主机能够通过 SSL 与被监测主机上的 NRPE 正常通信。

```
NRPE v2.12
```

但是如果出现一些 error 信息，则需要检查配置，检查的内容包括主要有下面几项：

1．nrpe 的版本号和 nrpe-plugin 的版本号是否一致。版本不一致极有可能造成该问题。

2．SSL 是否被关闭。确保 NRPE 以及 check_nrpe 插件在编译的时候都加入了 SSL 支持，同时在运行时都开启 SSL。不过一般编译过程中默认都会假如支持 SSL 选项。

3．确保 NRPE 的配置文件 nrpe.cfg 文件可以被 nagios 用户读取并且 nagios 用户可以执行 nrpe 二进制程序。

4．确认在/etc/xinetd.d/nrpe 文件的“only_from=x.x.x.x”中 x.x.x.x 是访问 NRPE 的监测主机的 IP 地址。

NRPE 的配置文件/usr/local/nagios/etc/nrpe.cfg 中实际上已经包含了一些对系统进行监测的命令。由于 NRPE 安装在本地，这些命令可以直接协助 NRPE 从被监测主机获取系统和服务运行状况，而且都是在刚才通过 nagios-plugin 安装的。

如果从监测主机上运行这些命令进行监测，一切正确可以看到下面的效果。那么不难看出其中的奥妙——实际上完全可以利用在/usr/local/nagios/libexec/中的各种脚本并添加各项参数定制自己的监测内容。

那么到此为止我们就完成了在远程被监测主机上安装和配置 RNPE 的任务。现在需要在监测主机，也即是 192.168.1.10 上面安装和配置 check_nrpe 插件。

步骤大概分为：

第一，安装 check_nrpe 插件；

第二，为使用 check_nrpe 插件建立 Nagios 命令定义；

第三，建立 Nagios host 以及服务定义

由于我们刚才已经在安装 Nagios 之后安装了 nrpe，所以实际上第一个步骤已经完成。

现在开始执行第二步骤——建立命令定义：

这里需要花点时间特别说明一下 Nagios 利用命令定义进行监测的原理：

在安装 nagios 成功之后可以看到在/usr/local/nagios/libexec 目录下有很多可执行监控程序或者脚本，其名称类似 check_icmp 这样的格式。Nagios 并没有提供针对这些监控程序的脚本的说明文档，想了解这些脚本如何工作，需要通过--h 参数，显示其使用方法和参数，并会给出一些实际的例子。例如：./check-disk –h。

那么我们可以尝试按照其中一个例子执行该脚本，执行和显示的结果如下：

./check_disk -w 10% -c 5% -p /tmp -p /var -C -w 100000 -c 50000 -p /dev/sda3

DISK OK - free space: / 2124 MB (41% inode=90%); | /=2955MB;4820;5088;0;5356

可以看到状态值“OK”，以及一些详细的数据信息。

上述操作说明这些插件都是可以独立使用，Nagios 有很多个 cfg 档用来定义各式各样的信息，其中 template.cfg 是用来定义主机和服务信息的模板文件，在目录/usr/local/nagios/etc/objects 下。我们按照这些模板来建立新的配置文件，例如同样目录下的 server.cfg 来定义监控内容，那么这些插件就会从 server.cfg 中调用。例如要定义一个需要监控的 SSH 服务，名称为 TestSSH：

按照其格式：

define service {

host_name x.x.x.x

service_description check_ssh

……

check_command check_ssh

}

host_name 项说明该服务所在的主机名，service_description 项为服务的说明信息，这项内容会显示在 nagios 页面中。check_command 项说明要使用的命令，这个例子中的命令 check_ssh 就是一个插件了。这个服务定义，明确了 nagios 在需要监控的内容和监控的手段，即使用 check_ssh 插件来监控主机 x.x.x.x 上的 ssh 服务情况。

除了直接使用插件来做 check_command 项的参数以外，还可以使用自己定义的命令来。例如，定义一个需要监控的主机，名字是 localhost.localdomain：

define host {

host_name localhost.localdomain

alias remotehost01

address 192.168.0.1

……

check_command check-host-alive

……

}

在此例中，check_command 项的参数“check-host-alive”并非一个插件，而是在 commands.cfg 档中定义的一个命令。那么在相同目录的 command.cfg 中对该命令又被定义为：

'check-host-alive' command definition

define command{

command_name check-host-alive

command_line $USER1$/check_ping -H 192.168.1.220 -w 300.0,80% -c 500.0,100% -p 1

}

首先，$USER1$这个参数在 resource.cfg 中定义，这个值会指向插件的目录（如：/usr/local/nagios/libexec）。“-H 192.168.1.220”定义目标主机的地址，-w 说明后面的一对值对应的是“WARNING”状态，“80%”是其临界值。“-c 500.0,100%” 其中“-c”说明后面的一对值对应的是" CRITICAL"，“100%”是其临界值。“-p 1”说明每次探测发送一个包。

所以归根结底就是说通过 ping 这种方式来证明某台主机处于 alive 状态。

而至于如何监听非默认埠的服务。下面我也举例说明一下这个问题：

例如：现需检查的一个运行在 8080 埠上的 http 服务。那么我们可以对 commands.cfg 档中对关于 check_http 的声明做如下修改。

```
# 'check_http' command definition
define command{
command_name check_http
command_line $USER1$/check_http -H 192.168.1.220 -p $ARG1$
}
```

其中$ARG1$是指在调用这个命令的时候，命令后面的第一个参数。

再把 services.cfg 中，对应服务的检测命令后面加一个参数：

```
define service {
host_name ...
...
check_command check_http!8080
}
```

这样就可以对 8080 埠的 http 服务进行监控了。如果要添加多个参数的时候，也可以类似操作。

综上，插件的安装和调用方法也就举例介绍完毕了，大家在使用中也可以使用自己写的检测脚本来完成比较特殊的检测功能。

所以按照上面所叙述的原理，我们开始第二步和第三步的配置，为使用 check_nrpe 插件建立 Nagios 命令定义以及服务定义：

修改配置文件/usr/local/nagios/etc/objects/command.cfg 并增加下面的内容：

```
define command{
command_name check_nrpe
command_line $USER1$/check_nrpe -H $HOSTADDRESS$ -c $ARG1$
}
```

然后针对要监测的目标主机建立主机定义，主机定义的项目和内容有很多，所以定义的项目如下：

host_name host_name　　# 简短的主机名称

alias alias　　# 别名，可以更详细的说明主机

address address　　# ip 地址，当然如果 DNS 服务可用也可以写名称。如果你不定义该值，nagios 将会用 host_name 去寻找主机。

parents host_names　　# 上一节点的名称，也就是指从 nagios 服务器到被监控主机之间经过的节点，可以是路由器、交换机、主机等等。这个节点也要定义并且要被 nagios 监控。

hostgroups　　# 简短的主机组名称

check_command　　# 检查命令的简短名称，如果此项留空，nagios 将不会去判断该主机是否 alive。

max_check_attempts # 当检查命令的返回值不是“OK”时，重试的次数

check_interval # 循环检查的间隔时间。

active_checks_enabled # 是否启用“active_checks”

passive_checks_enabled # 是否启用“passive_checks”，及“被动检查”

check_period # 检测时间段简短名称，此处只是名称，具体的时间段要写在其它的配置文件

obsess_over_host # 是否启用主机操作系统探测。

check_freshness # 是否启用 freshness 测试。freshness 测试是对于启用被动测试模式的主机而言的，其作用是定期检查该主机报告的状态信息，如果该状态信息已经过期，

freshness 将会强制作主机检查。

freshness_threshold # fressness 的临界值，单位为秒。 如果定义为 0，则为自动定义。

event_handler # 当主机发生状态改变时，采用的处理命令的简短的名字

（可以在 commands.cfg 中对其定义）

event_handler_enabled # 是否启用 event_handler

low_flap_threshold # 抖动的下限值。所谓抖动，主要定义了这样一种现象：在一段时间内，主机（或服务）的状态值频繁发生变化，类似一个问题风暴或者一个网络问题。

high_flap_threshold # 抖动的上限值

flap_detection_enabled # 是否启用抖动检测

process_perf_data # 是否启用 processing of performance data

retain_status_information # 程序重启时，是否保持主机状态相关的信息

retain_nonstatus_information # 程序重启时，是否保持主机状态无关的信息

contact_groups # 联系人组（这个组会在 contactgroup.cfg 文件中定义），在此组中的联系人都会受到该主机的告警提醒信息。

notification_interval # 告警临界值。达到此次数之后，才会发送该机的报警提醒信息。

notification_period # 告警时间段

notification_options # 告警包括的状态变化结果

notifications_enabled # 是否启用告警提醒功能

stalking_options [o,d,u] # 持续状态检测参数，o = 持续的 UP 状态，

d = 持续的 DOWN 状态，and u = 持续的 UNREACHABLE 状态.

当然在企业的监测环境中很多项目可能都不一定能够用上，这里我只是通过一个简单的例子说明其用法就好了。所以修改/usr/local/nagios/etc/objects/localhost.cfg 檔，在檔的“HOST DEFINITION”部分，在原来的基础上增加自己的主机定义内容：

define host{

```
use linux-box                   ; Inherit default values from a template
host_name localhost             ; The name we're giving to this server
alias RHEL5u2                   ; A longer name for the server
address 192.168.1.220           ; IP address of the server
}
```

同时在“HOST GROUP DEFINITION”部分，将 192.168.1.220 这台主机加入到 linux-servers 这个 hostgroup 中。如果有多台主机都属于这个 hostgroup，可以用逗号将其隔开。以下是我添加的内容：

```
define hostgroup{
    hostgroup_name  linux-servers ; The name of the hostgroup
    alias           Linux Servers ; Long name of the group
    members         192.168.1.220 ; Comma separated list of hosts that belong to this group
    }
```

而在最后的“SERVICE DEFINITION”部分，所有未注释的部分实际上是关于对 localhost 也就是本机所要监测的内容。其格式和语法就是我在提到 Nagios 检测原理方面举例说明的内容。对于 localhost 来说不需要修改了，但是可以把他的内容复制到自定义的 cfg 档中并照葫芦画瓢修改成对 192.168.1.220 这台主机的命令定义。我们下面就来做这样的操作：

在/usr/local/nagios/etc/objects 目录下针对监测的服务建立服务定义，建立一个新的文件 remotehosts.cfg，加入下面内容：

下面是自定义的：

```
define service{
use generic-service
host_name localhost
service_description CPU Load
check_command check_nrpe!check_load
}
```

表示监测远程主机的 CPU 负载。

如果要监测当前在远程主机的磁盘空间，则加入：

```
define service{
use generic-service
host_name localhost
```

service_description /dev/sda3 Free Space

check_command check_nrpe!check_disk /dev/sda3

}

如果要监测当前远程主机的僵死进程数，则加入：

define service{

use generic-service

host_name localhost

service_description Zombie Processes

check_command check_nrpe!check_zombie_procs

}

同时使用 vi 编辑器末行模式的 r 功能读取当前目录下的 localhost.cfg 档，删除“HOST DEFINITION”和“HOST GROUP DEFINITION”部分。只保留“SERVICE DEFINITION”部分并修改为下面的内容：

第一个命令定义：

通过 check_ping 脚本确保监测主机和被监测主机的连通性，如果丢包率到达 20%则报 warning，到达 60%则报 critical：

```
define service{
    use                 local-service       ; Name of service template to use
    host_name               192.168.1.220
    service_description             PING REMOTE HOST
    check_command           check_ping!100.0,20%!500.0,60%
    }
```

第二个命令定义：

监测远程主机根分区磁盘状况，如果可用空间低于 20%会报 Warning，如果可用空间低于 10%则报 Critical：

```
define service{
    use                 local-service       ; Name of service template to use
    host_name               192.168.1.220
    service_description         Root Partition of Remote Server
    check_command           check_local_disk!20%!10%!/
    }
```

第三个命令定义：

监测远程主机当前的登录用户数量，如果大于 20 用户则报 warning，如果大于 50 则报 critical：

```
define service{
    use                 local-service       ; Name of service template to use
    host_name               192.168.1.220
    service_description         Current Users of Remote Server
    check_command           check_local_users!20!50
    }
```

第四个命令定义：

监测远程主机当前的进程总数，如果大于 250 进程则报 warning，如果大于 400 进程则报 critical：

```
define service{
    use                 local-service       ; Name of service template to use
    host_name               192.168.1.220
    service_description           Total Processes of Remote Machine
    check_command           check_local_procs!250!400!RSZDT
    }
```

第五个命令定义：

监测远程主机当前的本地负载量：

```
define service{
    use                 local-service       ; Name of service template to use
    host_name               192.168.1.220
    service_description         Current Load of Remote Machine
    check_command           check_local_load!5.0,4.0,3.0!10.0,6.0,4.0
    }
```

第六个命令定义：

监测远程主机 swap 文件系统使用量，如果 swap 可用空间低于 20%则报 warning，低于 10%则报 critical：

```
define service{
    use                 local-service       ; Name of service template to use
    host_name               192.168.1.220
    service_description         Swap Usage of Remote Server
    check_command           check_local_swap!20!10
```

```
    }
```

第七个命令定义：

监测 SSH 连接可用性，但消息通知功能默认被关闭，因为并不是所有用户都有权限 SSH。

```
define service{
    use                     local-service       ; Name of service template to use
    host_name               192.168.1.220
    service_description         SSH of Remote Machine
    check_command           check_ssh
    notifications_enabled       0
    }


# Define a service to check HTTP on the remote machine.
# Disable notifications for this service by default, as not all users may have HTTP enabled.
```

第八个命令定义：

监测远程主机上的 HTTP 服务，但同 SSH，该服务的消息通知功能默认关闭。

```
define service{
    use                     local-service       ; Name of service template to use
    host_name               192.168.1.220
    service_description         HTTP of Remote Machine
    check_command           check_http
    notifications_enabled       0
    }
```

保存该档后，按照其它 cfg 文件的权限和属性为该文件指定所属用户和组：

```
# chown nagios.nagios /usr/local/nagios/etc/objects/remotehosts.cfg
```

至于想定义的其它内容，我就不再向该档中添加了，我想大家应该已经掌握了这种命令定义的方法了。

最后不要忘了一步关键的操作——在主配置文件中定义 Nagios 启动之后读取刚才修改的这些配置，也就是确保刚才修改的配置文件在 nagios 主配置文件/usr/local/nagios/etc/nagios.cfg 中都有正确指定，信息如下：

```
cfg_file=/usr/local/nagios/etc/objects/commands.cfg
cfg_file=/usr/local/nagios/etc/objects/remotehosts.cfg
```

cfg_file=/usr/local/nagios/etc/objects/localhost.cfg

最后校验配置文件正确性：

/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg

如果校验完全通过，则重启 Nagios 服务：

chkconfig --level 345 nagios on

service nagios restart

此时如果再通过浏览器访问 http://192.168.1.10/nagios，我们就可以看到被监测主机 192.168.1.220 上所反应出来的内容信息。

到此为止，Nagios 的基本原理和强大的功能就基本介绍完了。而事实上 Nagios 不但能在现有的功能基础上实现功能扩展，而且还能够实现和第三方软件的结合，例如和前面所介绍的 Mrtg 和 Cacti 联用来构建动态显示图标；同时按照前面所说明的内容可以配置各种事件级别的邮件通知功能。但因为篇幅的限制我们会在下次有机会的时候向大家介绍。尽管在配置的难度上显得比较高，但是其强大的功能和灵活性却给我们留下了极深的印象。因此这也是在一些中型甚至是大型企业中所推崇并逐步采用的一种监测方案。

PART3：各种性能监测优缺点比较和总结：

通过该文章的介绍，我们可以比较几种不同监测方法的特点。

在大多数监测环境中，SNMP 都是公用的标准和协议。所以除了 Nagios 之外，基本上所有的监测环境也是围绕 SNMP 协议部署。

通过 SNMP + 闭源商业软件的部署监测的方案：

拥有配置简单且功能也相对强大的优点，但是缺点是要受到闭源商业软件在功能和灵活性上的限制，而且意味着企业需要为这种类型的监测支付高昂的软件使用成本；

通过 SNMP + Mrtg 实现部署监测方案：

优点是费用方面的支出基本为零，但缺点是配置过程要显得复杂和繁琐，而且由于 MRTG 本身的一些限制功能比较单一。

通过 SNMP + Cacti + RRDtool 实现部署监测方案：

优点是同样不需要支付高昂的费用，而且相对于单纯使用 Mrtg 而言功能方面大大增强，显示的效果方面也要比 Mrtg 好很多，同时在监测内容和灵活性方面也有了很大的改善，相信这种方案能够被很多中小型企业所接受。但最大的问题是在部署的难度增加，对操作管理人员技术方面的要求也大大增加。因此这种方案也能够作为一种折中的选择。

通过 Nagios 实现监测方案：

这是几种不同方案中唯一可以不使用 SNMP 的，但是其丝毫不比 SNMP 协议基础上的监测功能逊色，甚至实现了更多实用的特性。同时部署和定义非常灵活，和其它软件的兼容性方面也表现出很多创造性的优势。显然在几种监测方案中，这种监测方案无疑是最优秀的！但缺点自然也不言而喻，强大的功能是以更为繁琐和更高的技术要求作为代价，如果针对一个大型网络要将 Nagios 所有的功能

都一一实现，显然对用户的技术水准方面要求会比较高。

总之，在企业系统和应用监测的领域中，尽管有各种不同类型的监测要求，尽管也相应地也提供了各种不同类型的监测部署方案。但不管是利用基本的 SNMP 实现简单和单一的监测，还是利用像 Cacti + RRDtool 甚至 Nagios 这样的软件实现功能更加强大的监测部署；不管是全部利用开源软件本身实现所有监测功能，还是和像 Whatsup 和 Solawins 等这种闭源商业软件结合部署监测环境——各种开源软件以及开源项目上都表现出了极强的共通性和兼容性，而且在功能上丝毫没有逊色。完全能够支撑和满足企业级的监测部署环境要求！

通过本文笔者希望能够为更多中小企业甚至大型企业用户在部署监测环境方面提供一些有用参考和帮助。希望他们能够藉助开源方案量体裁衣地打造企业级监测系统。

inotify + rsync 实现 linux 文件实时同步

ChinaUnix 网友：xjc2694

公司一套系统的同步使用的 donotify，不能实现子目录的实时同步，通过查资料，发现 inotify 可以实现子目录的实时同步，以下为笔记。

一、介绍

Inotify 是文件系统事件监控机制，作为 dnotify 的有效替代。dnotify 是较早内核支持的文件监控机制。Inotify 是一种强大的、细粒度的、异步的机制，它满足各种各样的文件监控需要，不仅限于安全和性能。

inotify 可以监视的文件系统事件包括：

IN_ACCESS，即文件被访问

IN_MODIFY，文件被 write

IN_ATTRIB，文件属性被修改，如 chmod、chown、touch 等

IN_CLOSE_WRITE，可写文件被 close

IN_CLOSE_NOWRITE，不可写文件被 close

IN_OPEN，文件被 open

IN_MOVED_FROM，文件被移走,如 mv

IN_MOVED_TO，文件被移来，如 mv、cp

IN_CREATE，创建新文件

IN_DELETE，文件被删除，如 rm

IN_DELETE_SELF，自删除，即一个可执行文件在执行时删除自己

IN_MOVE_SELF，自移动，即一个可执行文件在执行时移动自己

IN_UNMOUNT，宿主文件系统被 umount

IN_CLOSE，文件被关闭，等同于(IN_CLOSE_WRITE | IN_CLOSE_NOWRITE)

IN_MOVE，文件被移动，等同于(IN_MOVED_FROM | IN_MOVED_TO)

注：上面所说的文件也包括目录。

二、为能在 shell 下使用 inotify 特性，需要安装 inotify-tools

1、inotify-tools：The general purpose of this package is to allow inotify's features to be used from within shell scripts.

下载地址：http://inotify-tools.sourceforge.net/

编译安装

./configure

make

make install

完成后，注意查看 manpage，man inotify 、 man inotifywait

- inotifywait 仅执行阻塞，等待 inotify 事件。您可以监控任何一组文件和目录，或监控整个目录树（目录、子目录、子目录的子目录等等）。在 shell 脚本中使用 inotifywait。
- inotifywatch 收集关于被监视的文件系统的统计数据，包括每个 inotify 事件发生多少次。

2、inotify 的系统相关参数：

/proc interfaces

The following interfaces can be used to limit the amount of kernel memory consumed by inotify:

/proc/sys/fs/inotify/max_queued_events

The value in this file is used when an application calls inotify_init(2) to set an upper limit on the number of events that can be queued to the corresponding inotify instance. Events in excess of this limit are dropped, but an IN_Q_OVERFLOW event is always generated.

/proc/sys/fs/inotify/max_user_instances

This specifies an upper limit on the number of inotify instances that can be created per real user ID.

/proc/sys/fs/inotify/max_user_watches

This specifies a limit on the number of watches that can be associated with each inotify instance.

3、inotifywait 相关的参数（更多，查看 manpage）：

inotifywait

This command simply blocks for inotify events, making it appropriate for use in shell scripts. It can watch any set of files and directories, and can recursively watch entire directory trees.

-m, --monitor

Instead of exiting after receiving a single event, execute indefinitely. The default behaviour is to exit after the first event occurs.

-r, --recursive

Watch all subdirectories of any directories passed as arguments. Watches will be set up recursively to an unlimited depth. Symbolic links are not

traversed. Newly created subdirectories will also be watched.

-q, --quiet

If specified once, the program will be less verbose. Specifically, it will not state when it has completed establishing all inotify watches.

-e <event>, --event <event>

Listen for specific event(s) only. The events which can be listened for are listed in the EVENTS section. This option can be specified more than once. If omitted, all events are listened for. use “，” separate multi events

三、使用

1.查看是否支持 inotify，从 kernel 2.6.13 开始正式并入内核，RHEL5 已经支持。

看看是否有 /proc/sys/fs/inotify/目录，以确定内核是否支持 inotify

[root@RHEL5 Rsync]# ll /proc/sys/fs/inotify

total 0

-rw-r--r-- 1 root root 0 Oct 9 09:36 max_queued_events

-rw-r--r-- 1 root root 0 Oct 9 09:36 max_user_instances

-rw-r--r-- 1 root root 0 Oct 9 09:36 max_user_watches

2.关于递归:

inotifywait

This command simply blocks for inotify events, making it appropriate for use in shell scripts. It can watch any set of files and directories, and can recursively watch entire directory trees.

3.使用:

```
#!/bin/sh
src=/opt/webmail
des=/tmp
ip=192.168.7.192
/usr/local/bin/inotifywait -mrq --timefmt '%d/%m/%y %H:%M' --format '%T %w%f' \
 -e modify,delete,create,attrib \
${src} \
```

```
| while read  file
    do
        rsync -avz --delete --progress ${src} root@${ip}:${des} &&
        echo "${src} was rsynced"
        echo "-------------------------------------------------------------------------"
    done
```

注：当要排出同步某个目录时，为 rsync 添加--exculde=PATTERN 参数，注意，路径是相对路径。详细查看 man rsync

当要排除都某个目录的事件监控的处理时，为 inotifywait 添加--exclude 或--excludei 参数。详细查看 man inotifywait

另：/usr/local/bin/inotifywait -mrq --timefmt '%d/%m/%y %H:%M' --format '%T %w%f' \

-e modify,delete,create,attrib \

${src} \

上面的命令返回的值类似于：

10/03/09 15:31 /wwwpic/1

这 3 个返回值做为参数传给 read，关于此处，有人是这样写的：

inotifywait -mrq -e create,move,delete,modify $SRC | while read D E F;do 细化了返回值。

注：要取得监控文件发生的事件，在--format 处指定%e 参数，同时，使用--event 参数来指定要监控的事件即可，如--format '%T %w%f %e' --event modify,delete,create,attrib

说明： 当文件系统发现指定目录下有如上的条件的时候就触发相应的指令，是一种主动告之的而非我用循环比较目录下的文件的异动，该程序在运行时，更改目录内的文件时系统内核会发送一个信号，这个信号会触发运行 rsync 命令，这时会同步源目录和目标目录。

--timefmt：指定输出时的输出格式

 --format： '%T %w%f'指定输出的格式，上面的输出类似于：12/10/08 06:34 /opt/webmail/dovecot-1.1.2/src/test/1

小脚本，同步到多台主机：

文件: inotify_rsync.tar.gz

大小: 1KB

下载: 下载

更改后，更简单，适用于同步到相同的目录，监控多目录，多文件，同步到多台服务器

```
#!/bin/sh
#set -x
#var
src="/usr/local/nginx/html/lib /usr/local/nginx/html/www
/usr/local/nginx/html/var/www.work.com.conf.php"
des_ip="172.18.1.35 172.18.1.36 172.18.1.37 172.18.1.38"
#function
inotify_fun ()
{
/usr/local/bin/inotifywait -mrq --timefmt '%d/%m/%y-%H:%M' --format '%T %w%f' \
-e modify,delete,create,move $1 | while read time file
do
for ip in $des_ip
do
echo "`date +%Y%m%d-%T`: rsync -avzq --delete --progress $1 $ip:`dirname $1`"
rsync -avzq --delete --progress $1 $ip:`dirname $1`
echo
done
done
}
#main
for a in $src
do
inotify_fun $a &
done
```

参考：http://www.ibm.com/developerworks/cn/linux/l-ubuntu-inotify/index.html

关于减少 rsync 的遍历，未完：考虑到被监测的目录每次有一下时间时都会触发 rsync，modify, delete,create,move 每次 rsync 都会遍历源目录，当被监测目录内文件特别多时，会造成系统资源的严重消耗，所以，让 rsync 每次只同步修改的文件。因为，如果从监控目录 mv 走一个目录，那么 rsync 只会报告找不到你移走的目录而无法删除备份机的应该删除的目录。所以，对于删除这个事件，没有办法了，只能同步被删除文件或目录的上级目录了。将事件分为两部分，modify, create, move 事件，触发 rsync，只同步修改了的文件。delete 事件，同步被删除文件或目录的

上级目录（不能越过要同步的根目录）。

关于脚本内容的一些说明：

rsync.conf 里的目录格式一定要注意，没有最后的“/"

boot.sh

对于 rsync 命令的目标地址，是由两部分组成的：

1、rsync.conf 里的 dest

des=`grep '^dest' ${basedir}/rsync.conf | cut -d '=' -f 2 `

2、被修改了的文件的完全路径，去掉源目录部分，去掉被修改的文件的文件名。

mb=`echo $file | awk -F "/" '{NF=NF-1;OFS="/";print $0}' | sed "s#$src##g"`

boot.sh

##

先做个记录，脚本在 delete 部分还有问题

```
#!/bin/sh
#
basedir=/home/jason/Rsync
destNum=`grep -c '^dest' ${basedir}/rsync.conf`
src=`grep 'local directory=' ${basedir}/rsync.conf | cut -d '=' -f 2`
des=`grep '^dest' ${basedir}/rsync.conf | cut -d '=' -f 2 `
#
inotifywait -mrq --timefmt '%d/%m/%y %H:%M' --format '%T %w%f %e' \
 --event modify,create,move,delete ${src} | while read date time file event
      do
          echo $event
        for i in $des
        do
          case $event in
          MODIFY | CREATE | MOVE | MODIFY,ISDIR | CREATE,ISDIR | MODIFY,ISDIR)
              #echo $src
              no_src_root_file_name=`echo $file | sed "s#$src##g"`
              final_target_dest=$i$no_src_root_file_name
```

```
                echo rsync -avz --delete --progress $file $final_target_dest
                rsync -avz --delete --progress $file $final_target_dest
        ;;
        DELETE | DELETE,ISDIR)
                src_file_up_dir=`echo $file | awk -F"/" '{NF=NF-1;OFS="/";print $0}'`
                no_root_src_file_up_dir=`echo $file | awk -F"/" '{NF=NF-2;OFS="/";print $0}' | sed
"s#$src##g"`
                final_target_dest_up_dir=$i$no_root_src_file_up_dir
        echo    rsync -avz --delete --progress $src_file_up_dir $final_target_dest_up_dir
                rsync -avz --delete --progress $src_file_up_dir $final_target_dest_up_dir
        ;;
        esac
      done
    done
##################################################################################
################################
```

rsync.conf

local directory=/EBS/www/projects

#dest_here

dest=root@174.129.219.40:/EBS/www

20090723:

--format '%T %w%f'

其 中的%f 参数： %f When an event occurs within a directory, this will be replaced with the name of the File which caused he event to occur. Otherwise, this will be replaced with an empty string.

如果不加%f 参数，当一个文件夹里的文件发生变化时，不会输出文件名（需要自己 echo 脚本里 read 的变量），而时输出发生变化的文件的文件夹名，可以将这一点用于减少 rsync 遍历中的 delete 事件。

另外，%w 是用于输出发生变化的文件的。

%w This will be replaced with the name of the Watched file on which an event occurred.

基于 openvpn 的简单 VPN 服务器搭建

ChinaUnix 网友：victorchang

以下是根据外国流行的 OPENVPN 的 VPS 服务器提供商模式建立,也就是所谓的 VPS 服务器的建立方法.
根据我的调查,以及帮助一些客户维护的外国 VPS 服务器的经验所得..
我只介绍具体的建立步骤,具体的技术理论就不做详细的介绍了.
只要按我的教程一步一步的做,一会就建立最简单的 OPENVPN 服务器了.....
不是基于证书,也不是基于用户名和口令的验证....只需要一个密钥文件就可以了...

一 OPENVPN 服务器端的配置
1.下载 openvpn 软件,从 openvpn.net

```
download openvpn-2.0.9.tar.gz and lzo-2.00.tar.gz
```

2.安装软件
(1) install lzo-2.0

```
tar zxvf lzo-2.00.tar.gz
cd lzo-2.00
 ./configure
 make
 make check
 make test (run a full test)
 make install (when logged in as root)
```

(2) install openvpn

```
tar zxvf openvpn-2.0.9.tar.gz
cd openvpn-2.0.9
./configure
make&make install
```

openvpn 默认安装在/etc/openvpn 目录中

3.配置 openvpn

(1)建立你所需要的 OPENVPN 环境配置文件

```
cd /etc/openvpn
[root@bawwgt1 openvpn]# vi vpsville_vpn1.conf
dev tun
```

```
ifconfig 172.16.10.1 172.16.10.2
keepalive 10 60
proto tcp-server    ##这里一定要和服务器的协议一致:服务器端与客户端都是 UDP 或服务器端是 tcp-server 对应客户端 tcp-client
port 1119        ##此端口任意指定,需要客户端与服务器端一致敬
#user nobody
#group nogroup
persist-tun
persist-key
comp-lzo
verb 3
secret /etc/openvpn/secret.key
~
```

(2)生成所需要的密钥

[root@localhost openvpn]# cd /etc/openvpn

openvpn --genkey --secret secret.key (生成服务器端与客户端的密钥)

(3)这样在/etc/openvpn 中将会有以下文件.
vpsville_vpn1.conf secret.key

4.使 openvpn 能够开机自动启动,并启用 openvpn 的 nat 功能

```
vi /etc/rc.local
(加入以下内容)
echo "1" > /proc/sys/net/ipv4/ip_forward
iptables -t nat -A POSTROUTING -s 172.16.10.2/32 -o venet0 -j SNAT --to your vpn internet ip1
iptables -t nat -A POSTROUTING -s 172.16.11.2/32 -o venet0 -j SNAT --to your vpn internet ip2
/usr/local/sbin/openvpn /etc/openvpn/vpsville_vpn1.conf &
```

二 客户端配置

```
编辑你的客户端的 openvpn 配置文件:示例如下:

dev tun
remote 200.200.200.200    ##你的 openvpn 服务器 IP
ifconfig 172.16.10.2 172.16.10.1
keepalive 10 60
proto tcp-client              ##这里一定要和服务器的协议一致:服务器端与客户端都是 UDP 或服务器端是 tcp-server 对应客户端 tcp-client
```

```
port 1119
persist-tun
persist-key
comp-lzo
verb 3
secret C:\\Program\ Files\\UltraVPN\\config\\secret.key  ##将服务器生成的密钥 拷到些目录下,和你的客户端 OPENVPN 配置文件应该在一个目录中
redirect-gateway def1   ##这个选项是说客户端的所有的流量都经过 VPN 通道出去
```

三,最后一步进行测试.
不详细讲述了,我想大家都应该会测试...
有好的建议或想法请联系我.谢谢您的关注!

补充:secret.key 密钥文件内容的样例:

```
#
# 2048 bit OpenVPN static key
#
-----BEGIN OpenVPN Static key V1-----
e595847ad4899143d6ce76f086ea2ea0
cdbfb20555feff1b7790c645ce347ee9
7aed9892e74a6062ced20f32c8d5cae5
34fe6eff13afa7503fd703fdd149d558
b9c67dab6b793f22382317131fd89b0a
59db067c5bb4ad5399630025fec45d3e
993c7503660ee6932e46e217780d30b1
ead75c30b2f9fa9556b893aa3d62eb53
525212c70e5c76707c5f262ccaf5cc88
13b62b0db7e58b4f2c4106bfb3678bcb
447eaecb3b8ece93a137709cbc31d5dd
e25c26d86b69892fe31a950a9c5c8a43
eb088b0b21f8eee3ebde6ede916eaf7a
3162d4374a9656f82c7efbec9ab1e2e8
3470b2117e8bfd4c6ef9524e4a20ca5e
54d7e7955780f768358da71ef59b0fee
-----END OpenVPN Static key V1-----
```

开源数据库“五虎将”

ChinaUnix 网友：李晨光

“五虎将”一说最早出自三国平话,在《三国演义》中刘备麾下有五员猛将分别是关羽、张飞、赵云、马超、黄忠,后人惯称“五虎上将”。今天我为大家介绍的是 Linux 平台下功能强大的“开源数据库

五虎将”。当今社会，信息已经成为一种至关重要的资源。为此许多大型企业购进各种大型商业数据库，并基于它们展开一系列的应用，从而确保企业得以持续发展然而这些大型商业数据库，虽然功能强大，可是价格也非常昂贵。对于许多中小型企业来说，过于昂贵的数据库成本，是阻碍各种数据库解决方案进入企业的重要因素。那么是否存在一些数据库，它具备足够的功能，而价格又是中小型企业所能负担的呢？有，那就是开源数据库，开源数据库有着速度快、易用性好、支持 SQL、对网络的支持、可移植性、费用低等特点而且完全能满足中小企业需求，其中 Mysql 等开源数据库应用非常广泛，据最新数据统计显示，MySQL 的用户有 1200 万，谷歌、雅虎和亚马逊等许多互联网公司都是它的用户，这大大提高了 linux 的竞争力，下面我们分别细说这数据库五虎将的实力吧。

1.MySQL 5

作为当今最流行的开放源码数据库之一，MySQL 数据库为用户提供了一个相对简单的解决方案，适用于广泛的应用程序部署，能够降低用户的 TCO。MySQL 是一个多线程、结构化查询语言(SQL)数据库服务器。MySQL 的执行性能高，运行速度快，容易使用。

MySQL 包括以下几个关键优势：

◆ 可靠的性能和服务 MySQL 向公众提供所有数据库服务器软件的早期版本，都是利用开放源码进行为期几个月的测试之后才发布作为生产之用。

◆ 易于使用和部署 MySQL 的结构体系易于定制，运行速度快，其独特的多存储引擎结构为企业客户提供了灵活性， 为数据库管理系统带来紧致性和稳定性， 易于部署。

◆ 自由获得源码可以随时访问 MySQL 源代码，其策略确保了自由性，避免锁定某家公司或平台。

◆ 跨平台支持 MySQL 可用于 20 多种不同平台，包括主要的 Linux 系统、Mac OS X、Unix 和 Windows

◆ 可信赖的开发力量 MySQL 拥有大量的用户基础，也拥有高素质、有经验的开发团队。

◆ 满足企业需求 MySQL 结构体系简单易用，运行速度极快，能够处理企业数据库绝大多数的应用需求。

2008 年 12 月 8 日,Sun Microsystems 公司宣布，正式对外提供 MySQL 5.1 软件 —— 这是全球最受欢迎的开源数据库 MySQL 的一个极其重要的新版本。MySQL 5.1 GA 版现通过以下三种模式提供，以满足不同用户的各种特殊需求：

◆MySQL Community Server —— Sun 的 MySQL 数据库的免费开源版。这一 GPL 许可的全功能软件的目标用户是个人技术用户，他不需要商业支持或是享有优惠的机上服务。

◆MySQL Enterprise Server —— 它作为 MySQL Enterprise 订购的一部分来提供，它最可靠、最安全，提供的是 MySQL 数据库的最新版本，其目标用户是有法人的 IT 用户。该模式的订户每月可收到快速软件升级服务，每个季度可收到带有最新补丁程序的“服务包” —— 还能访问仅供预览的监测工具，享受全天候 7*24 的生产技术支持。

◆MySQL Embedded Server —— 这是 MySQL 软件的商业许可模式，让 ISV 和 OEM 将一个高速的、占用空间很小的数据库嵌入或打包到他们自己的产品中，而不需要免费的 GPL 许可。

2. PostgreSQL

PostgreSQL 是一个功能齐全、开放源码的对象一关系性数据库管理系统(ORDBMS)。目前，PostgreSQL 的稳定版本为 8.4 版，具有丰富的特性和商业级数据库管理系统的特质。这是一次向高质量大型数据库管理系统方向的飞跃。PostgreSQL 是很富特色的开源数据库管理系统，其特性覆盖 SQL-2/SQL-92 和 SQL-3/SQL-99。

◆ 丰富的数据类型 PostgreSQL 包括了丰富的数据类型支持，其中有些数据类型连商业数据库都不具备，比如 IP 类型和几何类型等。

◆ 功能全面 PostgreSQL 是全功能的开源软件数据库，全面支持事务、子查询、多版本并行控制系统和数据完整性检查等特性。

◆ 活跃的开发队伍 PostgreSQL 拥有一支活跃的开发队伍，在他们的努力下，PostgreSQL 的质量日益提高，增强了人们使用 PostgreSQL 的信心。

◆ 丰富的接口 PostgreSQL 支持几乎所有类型的数据库客户端接口。

◆ 支持多种平台 PostgreSQL 是目前支持平台最多的数据库管理系统之一，所支持的平台多达十几种，包括不同的系统和不同的硬件体系。

◆ 满足商用需求 PostgreSQL 的特性已经完全可以满足绝大部分用户的需要，胜任任何中上规模的应用业务，甚至可以支持生产数据库达 TB 级大小的数据量，已经逼近 32 位计算的极限。

◆ 强大的扩展能力 PostgreSQL 拥有强大的扩展能力，可以容易地扩展数据类型、内部函数和操作符等。

从技术角度来说，POStgreSQL 采用经典的 C/S(Client/Server)结构，即一个客户端对应一个服务器端守护进程的模式。这个守护进程分析客户端来的查询请求，生成规划树，进行数据检索，并最终把结果格式化输出后返回给客户端。为了便于客户端的程序编写，由数据库服务器提供统一的客户端 C 接口。不同的客户端接口都源自这个 C 接口， 比如
ODBC、JDBC、Python、Perl、Tcl、C/C++和 ESQL 等。

PostgreSQL 还欠缺的是一些高端数据库管理系统所需的特性，比如联机热备份、数据库集群、更优良的管理工具、更加自动化的系统优化功能和用以提高数据库性能的机制等。这些也是 PostgreSQL 正在不断努力的。

3.Ingres r3

CA 公司在 2004 年 11 月发布适用于 linux 的 Ingres r3 数据库软件。Ingres r3 按照 C A Trusted Open Source License(CATOSL，CA 可信开放源代码许可)授杖，取得此授权的人可以查看 Ingres r3 数据库的源代码，并免费下载该软件。CATosL 由通用公共许可衍生而来， 符合 Open Source Iniliative (OSI) 的要求。

Ingres r3 数据库平台的新功能如下：

◆ 高可用性

Ingres r3 包含集群软件，当集群配置中的一个数据库或服务器节点出现故障时，仍能保证服务的不问断性。在预防系统故障的同时，Ingres r3 还提供“缩放自如”的功能，让用户把众多低成本的服

务器连结起来． 以强化信息处理的性能。

◆ 可扩展性和可靠性

Ingres 通过并行查询处理将单个查询细分为多个组件，利用所有现有资源并行处理这些组件，从而提供可伸缩性能。同时，Ingres 支持 Oracle Cluster File System(OCFS)for Linux 和 IBM Distributed Lock Manager(OpenDLM)，为用户提供全新的群集功能，获得所需的可扩展性和可靠性。

◆ 技术与性能

Ingres 是第一个以 Zope RDBMS Persistence 引擎为基础的初始数据库(Initial Database)，其表分区和索引功能满足超大型数据库部署的需求。

◆ 集成性

Ingres 可以在异构环境中与其它应用程序和数据进行无缝集成。随着 Linux 在企业 IT 环境中的渐趋流行，这一集成功能尤为重要。其易于集成的特点使它能够与多种应用开发工具一起使用。此外，Ingres 使用行业标准的连接选件，支持开发人员在 J2EE 框架、．NET 环境，或者同时在两个环境下工作，特别适用于嵌入式应用。

◆ 服务

CA 公司将为 Ingres r3 提供支持和保障服务， 同时 CA 技术服务中心还提供多种可定制的培训课程和服务，包括现场培训或远程培训，这些培训和服务可以帮助客户更加有效地利用 Ingres r3 的特性。

4.MaxDB

MaxDB 前身是企业级的开源数据库 SAP DB，现由 MySQL 继续组织开发。MaxDB 是一个适应繁重任务、经过 SAP 认证的开源 OLTP 数据库，OLTP 的使用为其提供了可靠性、可用性、扩展性和高性能。MaxDB 拥有大型数据库的全面特点，与 Oracle 具有一定的兼容性，体积不大，可以在 Linux 上运行， 即将推出的 MaxDB 7.6 版本将支持 64 位计算技术，可以运行于 64 位的 Linux 平台和 HP-UX。

MaxDB 和 MySQL 这两个产品的外型相似。MySQL 的优势主要集中于产品的运行性能和稳定性，用户通过一个简单界面就可以容易地执行操作和管理。MaxDB 提供的先进性能则主要体现在企业级数据库的运用上。和 MySQL 相比，MaxDB 体型稍大，但与 Oracle、DB2 相比，几十兆的体积就能实现相近的功能，是相当 错的。目前，MaxDB 的各种管理器、查询器和客户端还在不断地完善中。

为吸引 Java 开发者，MaxDB 7.6 将支持由 IBM 创建的、基于 Java 的 Eclipse 开发框架。它还支持 MySQL 代理程序，允许 MaxDB 和 MySQL 产品共享数据，并允许开发者创建能透明使用这两种数据库的应用。MaxDB 还具备有监视性能，和能够自动提出保持平滑运行建议的工具。

MaxDB 适用于大型 mySAP Business Suite 环境，其它需要大型企业级数据库功能的场合， 以及用来补充 MySQL 数据库服务器的不足。高性能、可用性、运行的可靠性、可扩展性、易于使用，以及较低的总体成本正是企业部署 DBMS 环境所需要的若干特性。MaxDB 满足了企业用户的这些需求，其具体特性包括如下：

◆ 降低企业 SAP 运行的费用成本；

◆ 配置简单， 管理维护成本低廉；

◆ 完善的备份和恢复功能；

◆ 为大容量的用户和工作量而设计；

◆ 数据库容量可达 TB 级；

◆ 提供集群和热备份支持，带来高可用性；

◆ 同步管理器(Synchronization Manager)可以控制企业范围内的数据复制；

◆ 轻松使用图形化的数据库工具；

◆ 可用于所有的企业硬件和操作系统平台；

目前，全世界大约有 60000 名用户部署和应用 MaxDB 数据库，其中包 NToyota、Intel、DaimlerChrysler、Braun-Gillette、Bayer、Colgate、Yamaha 和 Deutsche Post (德国邮政局)等。作为一个目标指向企业级应用的开源数据库，MaxDB 正在不断地发展和完善中。

◆ 支持所有主要的 SAP 解决方案。

5.InterBase（即 Firebird）

InterBase 是一个易于开发者使用的数据库，可以支持复杂商业应用的快速开发与部署。同时，InterBase 也是一个友好、方便的商业数据库，可以提供支持关键性应用的企业级动力。Borland InterBase 7.5 是 InterBase 的最新稳定版本。它是一个高性能、跨平台数据晖，适合嵌人广泛部署的多用户应用中。

InterBase 7.5 的主要特性包括如下：

◆ 占用很少的空间意味着数据库消耗的系统资源很少，能够运行在一个并不昂贵的系统之上。

◆ 自动崩溃恢复功能自动崩溃恢复机制的调优功能使得系统维护量很小， 并且没有日志文件蔓延(Log Fi1eCreep)现象。

◆ 在线备份功能在线备份进一步降低系统维护量，并提升生产率，因为在备份数据时并不需要停止数据库。

◆ 安装简便简易的安装使得在没有 IT 支持人员的场合也能轻松部署，无需数据库管理员的参与。

◆ 快速．--r 靠地处理数据 InterBase 开创性地提出了活动数据库(Active Database)概念，把先进的自动化技术植入服务器内核。这些特性把数据处理步骤转移到服务器上，以得到更快和更可靠的运行。

◆ 极佳的速度与性能 InterBase 为支持关键性应用的嵌入式数据库提供了所需的速度和多用户性能。InterBase1]K 务器实现了多代体系结构(MGA)，可以同时对事务处理用户

和决策支持用户保证数据的高可用性。

◆ 降低开发费用 InterBase 可以帮助开发人员快速开发并部署应用， 从而降低开发费用。简单的安

装与较低的 Licesen 费用可以降低部署的费用。由于 InterBase 不需要数据库管理员的服务，所需的维护量非常少，后续的管理费用可以大大降低。

◆ 提升开发人员效率 InterBase 遵循 AQL92 标准，熟悉 SQL 标准数据库如 Oracle、IBM DB2 或 Sybase 的开发人员很容易就能对 InterBase 上手。InterBase 同时也与 Borland 的高效率开发环境紧密集成， 包括 Delphi、C++Builder 和 Kylix 等。

◆ 遵循工业标准以缩短开发周期 InterBase 与 ANSI/SQL、Java、Unicode，XML 和扩展数据表示(XDR)等工业标准保持严格的兼容，可以帮助开发人员降低开发、部署与维护跨平台应用所需的时间。

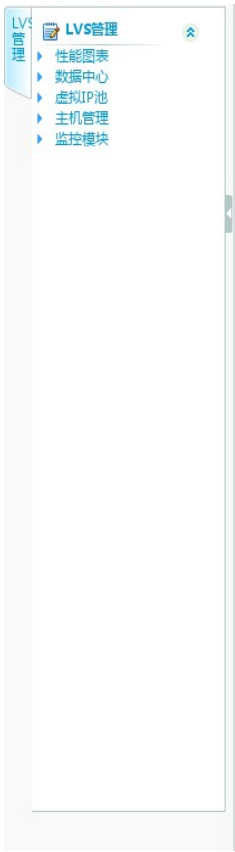
◆ 平台独立性 InterBase 可以部署在 Windows、Linux、Solaris 和其它 Unix 系统设备上，为一个平台所写的应用可以无需更改地运行在另一平台上，大大提高针对异构环境开发人员的生产率。

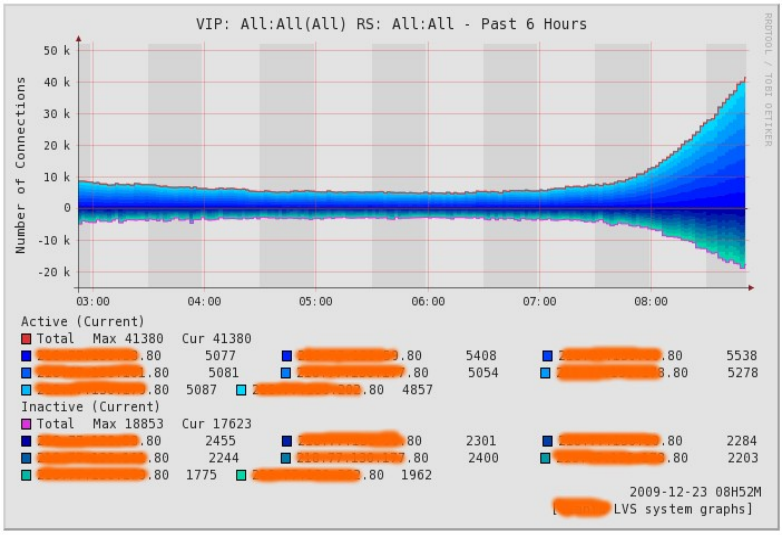
总结：我相信开源数据库使得信息处理的经济成本降低，将使强有力的数据库解决方案进入更多的企业，特别是中小型企业。随着开源数据库在功能上的日益强大和完善，以及人们对其了解的深入，Linux 开源数据库在中小型企业中将会有更广泛地应用。

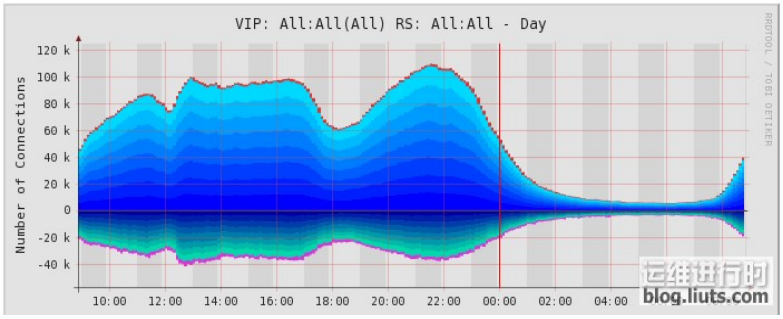
LVS 管理平台使用手册

ChinaUnix 网友：phpcool

自己开发的一套 LVS 管理平台，希望正在摸索当中的同志们有帮助，部份名词解释来之 LVS 官方。







为了更好管理、维护 LVS 平台，本人基于 Django+certmaster+func 开发了一套管理平台，主要功能模块分为性能图表、数据中心、虚拟 IP 池、主机管理、监控模块等功能，基本上是按 F5 – LTM 管理平台思路来设计，下面只要对这几大块功能进行说明。

1、性能图表

功能说明：以小时、日、星期、月、年的图表展示 LVS SERVER、VIP、SERVER 等流量情况，效果图如上：

2、数据中心

功能说明：管理 LVS 主机，一个 LVS 主机就是一个数据中心节点，功能包括新建、修改、生成、重载、图表、显示该组 VIP 等模块。

2.1 新建 LVS 节点：

全局定义块
1、 email 通知。作用：有故障，发邮件报警。这是可选项目，建议不用，用 nagios 全面监控代替之。
2、 Lvs 负载均衡器标识（lvs_id）。在一个网络内，它应该是唯一的。
3、 花括号“{}”。用来分隔定义块，因此必须成对出现。如果写漏了，keepalived 运行时，不会得到预期的结果。由于定义块内存在嵌套关系，因此很容易遗漏结尾处的花括号，这点要特别注意。

VRRP 定义块
1、同步 vrrp 组 vrrp_sync_group。作用：确定失败切换（FailOver）包含的路由实个数。即在有 2 个负载均衡器的场景，一旦某个负载均衡器失效，需要自动切换到另一个负载均衡器的实例是哪些?

2、实例组 group.至少包含一个 vrrp 实例。
3、Vrrp 实例 vrrp_instance.实例名出自实例组 group 所包含的那些名字。
（1）实例状态 state.只有 MASTER 和 BACKUP 两种状态，并且需要大写这些单词。其中 MASTER 为工作状态，BACKUP 为备用状态。当 MASTER 所在的服务器失效时，BACKUP 所在的系统会自动把它的状态有 BACKUP 变换成 MASTER；当失效的 MASTER 所在的系统恢复时，BACKUP 从 MASTER 恢复到 BACKUP 状态。
（2）通信接口 interface。对外提供服务的网络接口，如 eth0,eth1.当前主流的服务器都有 2 个或 2 个以上的接口，在选择服务接口时，一定要核实清楚。
（3）lvs_sync_daemon_inteface。 负载均衡器之间的监控接口，类似于 HA HeartBeat 的心跳线。但它的机制优于 Heartbeat，因为它没有“裂脑”这个问题，它是以优先级这个机制来规避这个麻烦的。在 DR 模式中，lvs_sync_daemon_inteface 与服务接口 interface 使用同一个网络接口。
（4）虚拟路由标识 virtual_router_id.这个标识是一个数字，并且同一个 vrrp 实例使用唯一的标识。即同一个 vrrp_stance,MASTER 和 BACKUP 的 virtual_router_id 是一致的，同时在整个 vrrp 内是唯一的。
（5）优先级 priority.这是一个数字，数值愈大，优先级越高。在同一个 vrrp_instance 里，MASTER 的优先级高于 BACKUP。若 MASTER 的 priority 值为 150，那么 BACKUP 的 priority 只能是 140 或更小的数值。
（6）同步通知间隔 advert_int .MASTER 与 BACKUP 负载均衡器之间同步检查的时间间隔，单位为秒。
（7）验证 authentication。包含验证类型和验证密码。类型主要有 PASS、AH 两种，通常使用的类型为 PASS，据说 AH 使用时有问题。验证密码为明文，同一 vrrp 实例 MASTER 与 BACKUP 使用相同的密码才能正常通信。

4、 虚拟 ip 地址 virtual_ipaddress . 可以有多个地址，每个地址占一行，不需要指定子网掩码。注意：这个 ip 必须与我们在 lvs 客户端设定的 vip 相一致!


2.2、 生成
功能说明：在管理端生成配置文件并校验。

2.3、  重载

功能说明：将配置文件传输到 LVS 节点服务器并重启 Keepalived 服务。

2.3、  图表

功能说明：显示该 LVS 节点流量数据。

3、虚拟 IP 池(VIP)

功能说明：添加、修改 VIP 池。

3.1、添加图示：

虚拟服务器 virtual_server 定义块

虚拟服务器定义是 keepalived 框架最重要的项目了，是 keepalived.conf 必不可少的部分。

1、 虚拟服务器 virtual_server. 这个 ip 来自于 vrrp 定义块的第“4”步，后面一个空格，然后加上端口号。定义一个 vip，可以实现多个 tcp 端口的负载均衡功能。

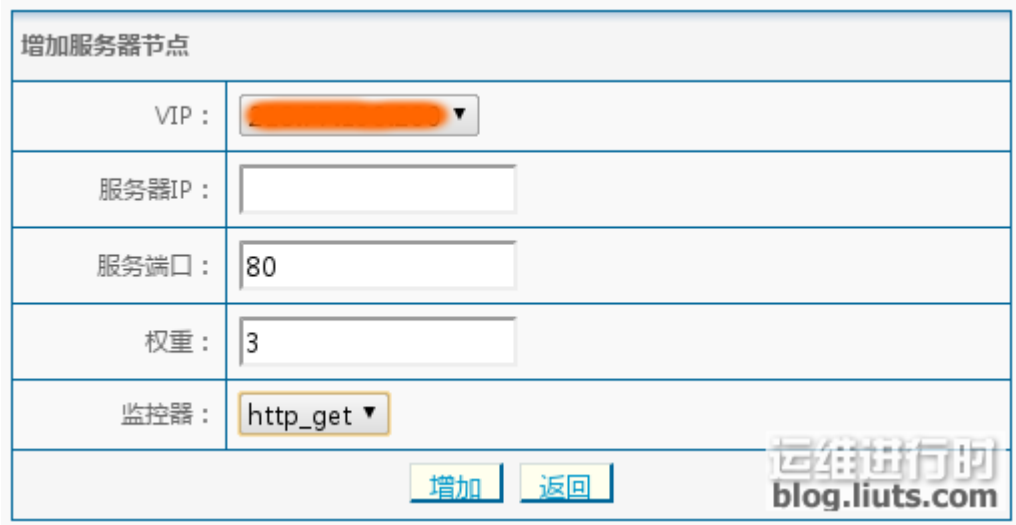
（1）delay_loop。健康检查时间间隔，单位是秒。
（2）lb_algo. 负载均衡调度算法，互联网应用常使用 wlc 或 rr。
（3）lb_kind. 负载均衡转发规则。一般包括 DR,NAT,TUN3 种，在我的方案中，都使用 DR 的方式。

（4）persistence_timeout.会话保持时间，单位是秒。这个选项对动态网站很有用处：当用户从远程用帐号进 行登陆网站时，有了这个会话保持功能，就能把用户的请求转发给同一个应用服务器。在这里，我们来做一个假设，假定现在有一个 lvs 环境，使用 DR 转发模式，真实服务器有 3 个，负载均衡器不启用会话保持功能。当用户第一次访问的时候，他的访问请求被负载均衡器转给某个真实服务器，这样他看到一个登陆页面，第一次访问完毕；接 着他在登陆框填写用户名和密码，然后提交；这时候，问题就可能出现了—登陆不能成功。因为没有会话保持，负载均衡器可能会把第 2 次的请求转发到其他的服务器。
（5）转发协议 protocol.一般有 tcp 和 udp 两种。实话说，我还没尝试过 udp 协议类的转发。

4、主机管理

功能说明：包括添加、修改、设置 LB 等功能模块。

4.1、添加 SERVER



真实服务器 real_server.也即服务器池。Real_server 的值包括 ip 地址和端口号。多个连续的真实 ip，转发的端口相同，是不是可以以范围表示？需要进一步实验。如写成 real_server 61.135.20.1-10 80 .

（1）权重 weight.权重值是一个数字，数值越大，权重越高。使用不同的权重值的目的在于为不同性能的机器分配不同的负载，性能较好的机器，负载分担大些；反之，性能差的机器，则分担较少

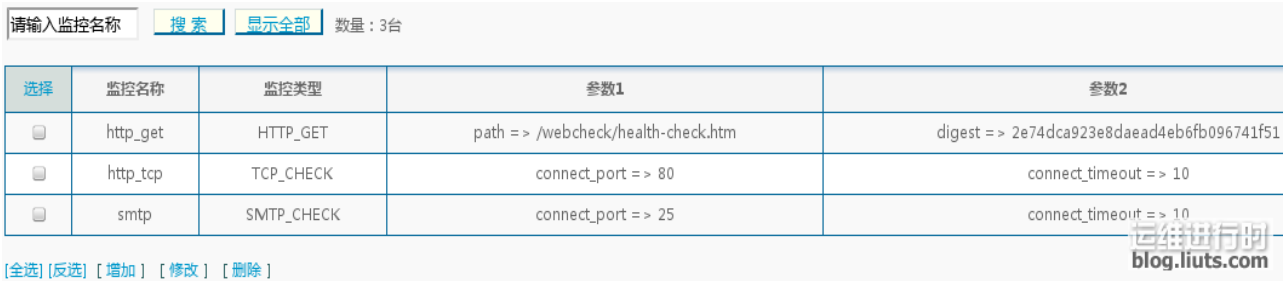
的负载，这样就可以合理的利用不同性能的机器资源。

4.2、 设置 LB

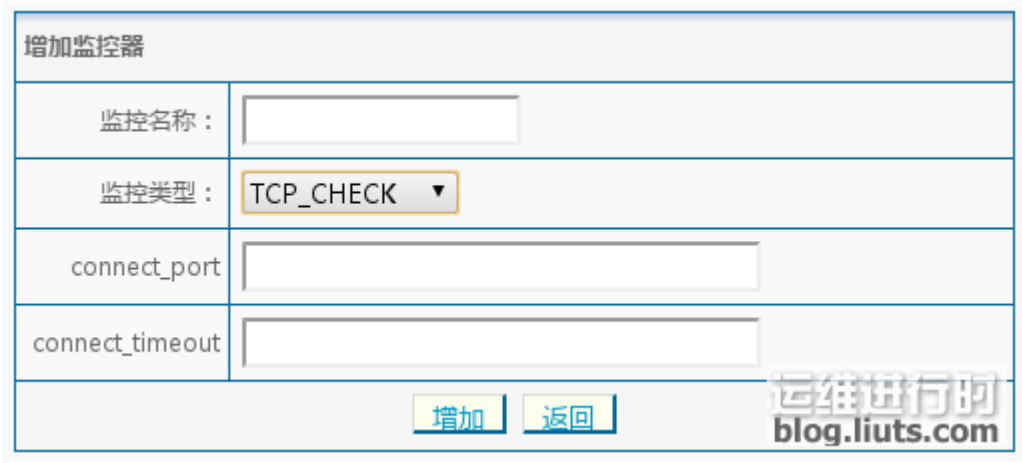
功能说明：设置所选服务器的 LoopbackIP，只限于 DR 模式。

5、监控模块

功能说明：添加、删除、修改自定义监控名称。



5.1、添加监控点



监控类型包括 SSL_GET、HTTP_GET、TCP_CHECK、SMTP_CHECK、MISC_CHECK 等，选择不同的监控类型会提示你输入不同的参数值，一般情况下选择管理员添加的类型即可满足，一些特殊应用的监控可以自定义去添加。

网友热评

****

热点技术评论

几种移动版操作系统的感受

unix 跟 linux 差别真 tmd 大

两层防火墙后面的 ftp 服务器问题

linux 上 netbackup 安装问题

内存问题请教

全局变量的地址为什么会改变?

c 语言太诡异了

awk 如何删除某一个域的换行符

指针不能相乘?

SQUID 运行时 CPU 利用率高

请教个文件修改存储方面的问题

文件和成员概念问题

跨越三层的远程镜像实例

legato 通过 vcb 备份恢复 vm 安装配置文档

mfs（mooseFS）文件系统

发布一个 mmap 的 trie

关于！强制保存文件的后果

dev_queue_xmit 导致死机

添加 storage 服务器的时候不能同步

求 telnet 的具体用法

有没有软件实现的 cache

如何消除 linux 启动时的信息

NAT 连接数达 2 万时数据丢包严重

RHEL4.7 无法加 swap

请教一下关于传送数据

热点新闻评论

数学是知识，哲学是智慧

目前 IT 就业前景问题

最失败的一次工作经历

sun 工程师出路在何方

linux 安全的有哪些认证

几个 PHP 面试难题分享

惠普发布中国黑客威胁论

架构真的比算法难多了

讨论一个服务器的设计方案

BSD 和 Linux 哪个好用?

AMD 和 Intel CPU 的 cflags 参数

大家 show 一下自己管理机器

ubuntu 当 server 为啥不被看好?

雨林木风欲借 Linux 谋划 3 年内上市

/BOOT 分区被格了

32 位 Linux 内核到底能识别多大的内存

迷茫了 linux

实在太烦袁萌这个人了！

CU 的 WEB 安全性有待加强

CU 技术图书活动规则改进和意见征集！

服务器中木马怎么办啊?

BSD 和 Linux 哪个好用?

RHEL 5.2 网卡启用的问题

感动 09，展望 10

剖析一个由 sendfile 引发的 linux 内核 BUG